PART I

CONVERTERS AT WORK (WITH AND WITHOUT MICROPROCESSORS)

Chapter One

Introduction—
Data Systems and Components

The analog-to-digital and digital-to-analog converter are key elements of any system that uses digital techniques to process or communicate analog "real-world" electrical data. This book is basically about a/d and d/a converter circuits: understanding them, applying them, testing them, choosing them, and using them in systems.

When used in systems, they are often accompanied by a variety of other devices, both analog and digital, to measure input signals and perform intermediate processing with varying degrees of sophistication.

Depending on how eager the user is to wrestle with the details of electronic circuitry, converters for data-acquisition may be purchased within a wide variety of forms. They range from simple integrated-circuit conversion chips to systems that accept an electrical sensor's output, provide processing programmed by the user, and generate appropriate analog or digital output signals to control physical variables.

This chapter provides a brief thumbnail sketch of elements that tend to be used in systems with converters and are likely to be found in block diagrams in this book. General characteristics and aptitudes are summarized, and their roles in relation to converters are hinted at within the short discussions devoted to each entity.

Analog vs. Digital

As used here, "digital" refers both to electrical signals that represent numbers, control logic, and physical variables that can be measured by counting or identifying discrete states—and to related circuitry. Examples include

event counts and binary* voltage.

"Analog" has to do with physical variables that are represented or measured by continuously variable aggregates, and to related circuitry. Temperature and electric current (as aggregates of molecular and electronic motion) and measurements of continuous quantities with linear scales (analogs) are examples of analog quantities.

In this book, by far the greater weight of discussion is given to the properties of *analog* circuits, in the performance of system functions. The reason for this is no mystery: the challenges to the analog circuit designer's ability are many, varied, and unrelenting.

The basic promise of analog circuits, in favorable environments, comprises functional simplicity (inherent parallel operation), high speed, and overall low cost, as well as the ability to mimic natural phenomena with electrical variables and parameters. The difficulties in dealing with analog circuits are a natural function of both the wide dynamic range that accompanies their extreme fine structure and the many degrees of freedom of interaction associated with them. In both concept and practice, designers must be concerned that analog circuits have to labor in the real world, where limits to resolution and accuracy are directly related to physical environment, electrical interference, signal magnitude, component tolerances, and the passage of time; and bandwidth adds one more dimension of complexity, being affected by all of the above.

Digital circuits, on the other hand, in dealing with binary quantities, have high (but by no means infinite!) noise immunity, no drift, high speed and low cost (individually); and the rules for using them are few and simple. With digital techniques, the principal challenges relate to the reduction of overall cost and complexity. They require ingenuity in pursuing optimal tradeoffs in the development of system architecture, avoidance of timing errors, the writing of foolproof software, and avoidance of problems arising from the nature of electronic parameters and circuitry—which are inherently analog!

Hence, designers of both analog and digital circuits, as well as software, must always anticipate where Murphy's Law will strike next and be prepared to debug when anticipation has failed.

With the exception of preamplification, a great many of the functions presented here in analog form may be performed digitally, after conversion. The choice of technology depends on tradeoffs of cost, speed, complexity, and requirements for adjustment and calibration. There has been a rapid increase, with no sign of a letup, in the the development of devices and subsystems that are intended to perform analog functions using digital components. Examples of these include analog function generation with read-only memories (ROM

*"Binary", in digital technology, has two meanings: *two-valued* (e.g., 1 or 0) and *base-2* (number system). The meaning is usually clear from the context.

lookup tables) and d/a converters, and the universe of arithmetic, logical, and control possibilities with microprocessors—including special-purpose μPs with analog capabilities and digital signal processors.

Figure 1.1 illustrates the relationships of the principal components of a data system in a "global" perspective; causality flows clockwise but can loop recursively. As a sampling of entities found in the loop, those elements or problems to be discussed in this chapter include:

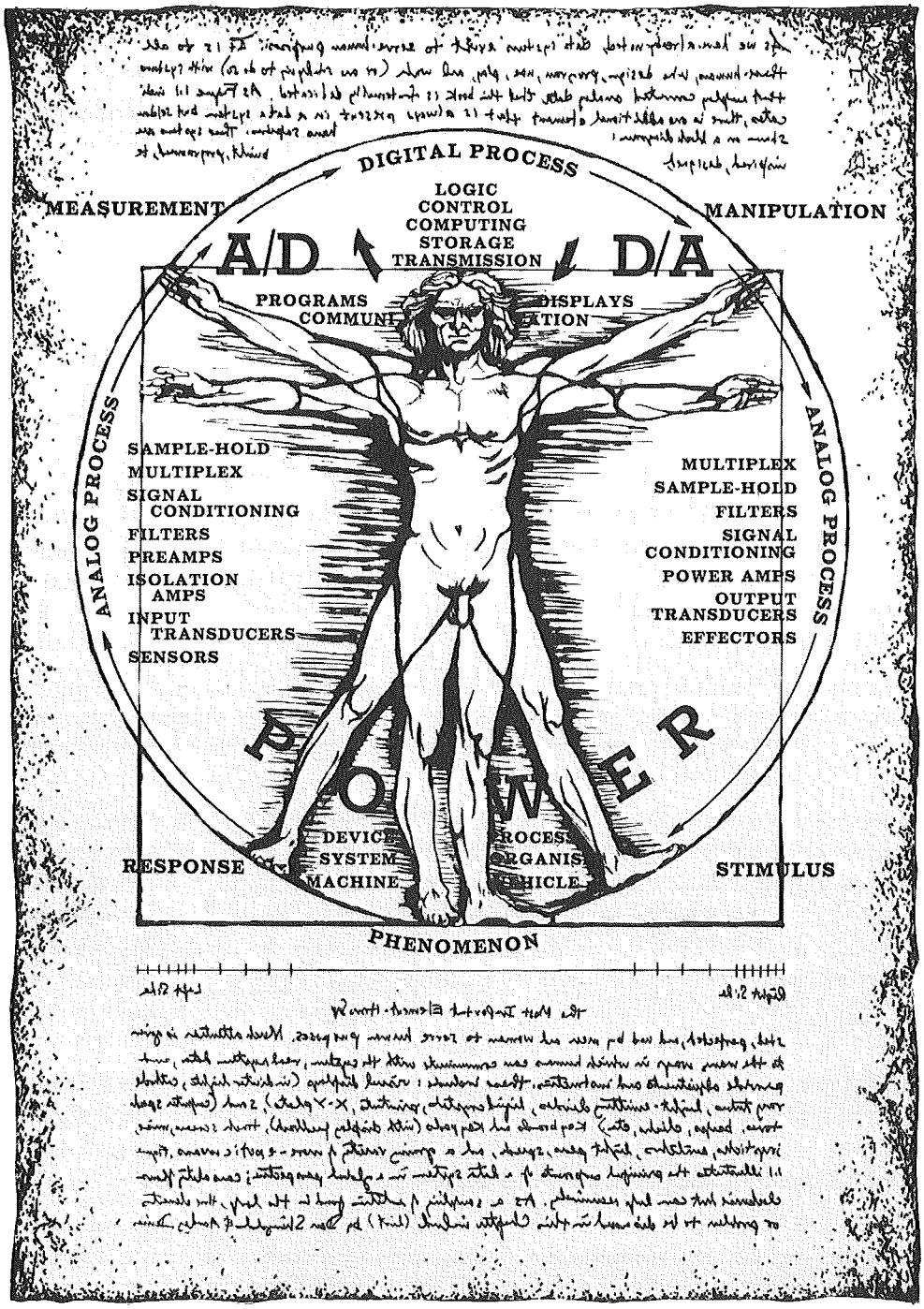
Sensors
Operational Amplifiers
Instrumentation Amplifiers
Isolators
Analog Function Circuits
Analog Multiplexers
Digital Multiplexing
Sample-Hold Circuits
Analog-to-Digital Conversion
Digital-to-Analog Converters
Registers
Microprocessors
Counters
Filters
Comparators
Power Supplies
Digital Panel Instruments

Rarely will a system involve all of the above elements; but most systems will use many of them—and others, not mentioned here, as well. The shrinkage in size and cost of components—or increase of the level of system integration—has made possible the combination of many of these elements into integral parts of subsystems, in the form of chips, modules, or boards, having specified performance and even a modicum of intelligence.

THE MOST IMPORTANT ELEMENT

As Figure 1.1 indicates, there is one additional element that is always present in a data system but seldom shown on a block diagram: *homo sapiens*. These systems are inspired, designed, built, programmed, tested, perfected, and used by men and women to serve human purposes.

Much attention is given to the many ways in which humans can communicate with the system, read system data, and provide adjustments and instructions. These include: visual displays (indicator lights, cathode-ray tubes, light-emitting diodes, liquid crystals, printouts, X-Y plots), sound (computer speech, tones, beeps, clicks, etc.), keyboards and keypads (with display feedback), touch screens, mice, joysticks, switches, light pens, speech, and a growing variety of more-exotic means.



(D. H. Sheingold, with apologies to Leonardo da Vinci: Rule of Proportions, Academy of Fine Arts, Venice)

Figure 1.1 Relationship of functions and causality in a data system.

SENSORS

One might imagine that the electronic systems designer has very little say in the choice of sensor, that (s)he accepts whatever data signals exist without protest and gets on with the interface system design without further ado. However, the systems engineer who can have a say in the selection of the original transducer can go a long way towards easing the conversion-design task.

For example, in monitoring or controlling mechanical shaft rotation, the designer may be confronted with signals obtained by three radically different position-sensing approaches: optical shaft encoders, resolvers, and potentiometers, plus variations on all three. For a given task, different sensors with appropriate speed, accuracy, and reliability specifications will result in greatly differing interfaces.

Similarly, temperature measurements may be accomplished with thermocouples, RTDs (resistance temperature-detectors), thermistors, or semiconductor temperature sensors; while mechanical force may be measured directly by load cells and strain gages, or obtained indirectly by integrating the output from accelerometers, or even by counting interference fringes in an optical system.

Although it is not our rôle to recommend any particular type of transducer for a particular application, we thoroughly endorse the idea of getting the electronic systems-design engineer into the act before the signal sources are decided upon, instead of later, when it may be found that the designer is painted into a corner by the few options allowed.

OPERATIONAL AMPLIFIERS

Even if the transducer signals must simply be converted from current to voltage, or scaled up from millivolt levels to an a/d converter's 5- or 10-volt full-scale input range, signal conditioning is required.[1] Because of their low cost, one or more operational amplifiers in a suitable circuit with appropriate closed-loop gains and additive constant offsets may be the first (but not necessarily the best) choice. If the system involves many analog sources, a choice must be made between providing each transducer with its own signal conditioning and a central signal-conditioning facility that can handle a number of multiplexed inputs.

Besides gain scaling, operational amplifiers are used for a host of mathematical and signal processing functions: linear and nonlinear, static and dynamic. Although op amps are cheap, such dedicated devices as instrumentation and isolation amplifiers are always worthy of consideration in specialized applications; the design time they save often makes them the most cost-effective option.

[1]Considerable useful information about signal conditioning may be found in the Analog Devices *Transducer Interfacing Handbook* (1980). Consult the Bibliography for details.

INSTRUMENTATION AMPLIFIERS

If analog data must be transmitted over long distances (or often, even over quite short distances), differences in ground potential between signal site and data center will add spice to the interface system design problem. In order to separate common-mode interference from the signal to be recorded or processed, devices designed for the purpose (for example, instrumentation amplifiers) may be used.

Instrumentation amplifiers are functionally complete components characterized by good common-mode-rejection, high input impedance, low drift, and adjustable gain. IC types compete favorably with operational-amplifier circuit kludges in cost, as well as size and performance. Today's instrumentation amplifiers are generally monolithic ICs; gains are programmable in some by external resistors, others contain internal precision resistors and are programmed by jumpers or software.

ISOLATORS

In the event of high common-mode voltage levels or the need for extremely low common-mode leakage current, or both (as might be mandatory for many clinical applications in medical electronics), galvanic isolation is required to interpose a break in the common-mode path from the analog signal source to the data system. *Isolation amplifiers* may involve optical isolation or (more often) transformer-coupled carrier techniques; they usually have at least 1,000 volts of isolation, and they typically cost more than instrumentation amplifiers. Though most often used for isolating input data from system level, they may also be used for communicating system outputs to destinations at high common-mode voltage.

When isolation is necessary in the digital world, digital logic signals and voltage-to-frequency-converter output pulse trains are usually isolated by solid-state opto-couplers or fiber optics.

Amplifiers and analog signal conditioners are not the only analog devices that are isolated. For example, many line-powered digital panel instruments have isolation between analog inputs and digital output. A typical isolated digital-to-analog converter isolates digital equipment from its analog output (a 4-to-20mA current loop) and power supply. Thus, it can protect the analog output signal by preserving the information stored in its registers during system maintenance or crashes, as well as permitting manual updating.

ANALOG FUNCTION CIRCUITS

These "analog-to-analog" converters are analog computational circuits and special-purpose devices used to condition analog signals. Where their accuracy is adequate, they can simply, at low cost and with high speed, relieve a

processor of an expensive and time-consuming software and computational burden.

The membership of this category is open-ended. Some of the more popular operations performed are multiplication; taking ratios; raising to powers; taking roots; performing special-purpose nonlinear functions such as linearizing transducers; performing rms measurements; computing trigonometric functions and vector sums; integrating and differentiating; transforming current to voltage or voltage to current, etc.

Some of these operations can be purchased in the form of such readily available devices as multiplier/dividers, log/antilog amplifiers, etc. Others represent but a sampling of the vast analog parallel number-crunching potential and program memory inherent in operational-amplifier circuitry, available to the competent designer at low parts cost, and limited only by human ingenuity.[2]

ANALOG MULTIPLEXERS

If data from many signal sources must be processed by the same computer or communications channel, via a single converter, a multiplexer is usually introduced to couple the input signals into the a/d converter in some preset or random sequence. An n-bit logic address input (2^n channels) determines which data source is to be coupled to the converter at any instant.

Multiplexers are also used in reverse, as distributors, or demultiplexers. For example, when the converter must distribute analog information to many different channels, the multiplexer, fed by a high-speed output d/a converter, can continually refresh the various output channels with updated information; generally, each channel must have analog storage to retain its information until the next update.

DIGITAL MULTIPLEXING

Digital systems often can do without a device that is specifically labeled "multiplexer" for parallel data. Such a device is cumbersome, requiring essentially a set of multipoint switches, one switch for each wire of the bus (16 multipoint switches for a 16-bit data bus) and a large number of wires converging on a single device. Instead, the digital multiplexing function is usually delegated to the devices being multiplexed, as they share a common input/output bus. They are connected to it via internal "three-state" switches.

When enabled, the switches connect a parallel set of individual 1's and 0's to the bus; otherwise, the data from the device is in effect disconnected (hence three states: 1, 0 or disconnected *). Addressed *read* commands from the pro-

[2]Considerable useful information about analog functional operations can be found in the *Nonlinear Circuits Handbook*. Consult the Bibliography for details.

*Three-state is a bit of a misnomer. Actually, there are two control states, *enabled* and *not-enabled*, and two data states, 1 and 0. The net result is three tangible states: 1, 0, and open.

cessor instruct the individual sources which one (and only one) among them must feed its burden of data onto the common bus, thence to its destination.

Input registers of all devices *receiving* data from the bus are connected to the bus. The device(s) chosen to receive the data appearing on the bus at a given time are strobed by a *write* signal, which latches the data into their registers.

SAMPLE/TRACK-HOLD CIRCUITS

In many interface systems, the analog signal varies quite rapidly. Since conversions take place at discrete (sampling) intervals, and an a/d converter cannot digitize the input signal instantaneously, substantial changes of the signal level during the actual conversion process could result in gross errors. The problem with the converter is that the conversion is completed at some appreciable (and not always constant) time following the repetitive, accurately timed conversion command, so that the final digital value would never truly represent the data level prevailing at the instant at which the conversion command was transmitted unless the analog signal was frozen at that instant.

Sample-hold devices make a fast acquisition of the varying signal, on a "sample" command and then—on a "hold" command—hold the signal constant at the output for the duration of the conversion process. Sample-hold circuits may also be used in multi-channel distribution installations, where they enable each channel to receive and hold its own signal level for activation of differing output processes. They are also used in "deglitching" to hold an output voltage steady while a large input transient is occurring during a d/a converter update, then quickly acquire the new data when the transient has subsided.

Typically, a sample-hold circuit used in data acquisition must acquire a signal rapidly (usually within microseconds), respond to the *hold* instruction within a fraction of a microsecond, with an uncertainty of a nanosecond or less, and hold the last value without significant "droop" for tens of microseconds. Most sample-holds also function as (and are often called) *track-holds*, i.e., once the analog signal is acquired, it is tracked until the hold command is received.

A/D CONVERTERS

These devices, which range from monolithic ICs to high-performance hybrid circuits, modules, and even boxes (such as digital panel meters), convert analog data—usually voltage—into an equivalent digital form. Key characteristics of a/d converters include absolute and relative accuracy, linearity, no-missing-codes, resolution, conversion speed, stability, and—of course—price. Other aspects open to choice include input ranges, digital output codes, interfacing techniques, presence of on-board multiplexing, signal conditioning, and memory.

Although the industry tends to converge upon the successive-approximations technique for a very large number of system applications, because of its inhe-

rently excellent compromise between speed and accuracy, other popular alternatives include integrating techniques (dual-ramp, quad-slope, and voltage-to-frequency), counting and tracking techniques (counter-comparator), and, for video-signal speeds, "flash" and digitally corrected subranging techniques.

Voltage-to-frequency converters can provide high-resolution conversion and such special features as long-term integration (from seconds to years), digital-to-frequency conversion, (with a d/a converter), frequency modulation, voltage isolation, and arbitrary frequency division or multiplication. Synchro-, resolver-, and Inductosyn® to digital converters are used where angular or linear position must be measured precisely and with high resolution, and converted into digital form.

A technique employed by some microprocessor users, to avoid the need for purchasing an a/d converter as such, is to use a d/a converter, a comparator, and the processor's logic to perform a tracking or successive-approximation conversion. While the financial cost is small, the software and program time burden is substantial.

D/A CONVERTERS

These devices reconstitute the original data after processing, storage, or even simple digital transmission from one location to another. The basic converter usually consists of an arrangement of weighted resistance values (or resistive or capacitive divider ratios), each controlled by a particular level or "significance" of digital input data, that is switched to develop varying output voltages, currents, or gains by selective summation in accordance with the digital input code.

The output of a d/a converter is proportional to the reference source used. Although most converters for data-handling applications are used with essentially fixed references, there is a special class of converter, capable of handling variable—and even bipolar ac—reference sources. These devices are termed *multiplying DACs*, because their output is the product of two variables—the number represented by the digital input code and the analog reference voltage; both may vary from full scale to zero, and even negatively.

Another way of looking at a multiplying DAC is to think of it as a digitally adjustable gain control. Some even have logarithmic conversion relationships, to permit digital inputs in decibels to control gain in steps having equal ratios.

For position outputs, the d/a converter may take the form of a digital-to-synchro or digital-to-resolver converter. And if the digital signal is a train of pulses at a given rate, it can be converted to analog by a frequency-to-voltage

®Trademark of Farrand Controls, Inc.

conversion circuit, often employing a voltage-to-frequency converter in a phase-locked loop.

REGISTERS

The digital register is a key component of digital systems. In this book's context, they are used to to hold information in readiness for passing it along from converters to computers, from computer buses to d/a converters, etc.

For example, a multi-channel interface system using an a/d converter for every input channel would store the parallel digitized values in an output register associated with each converter until called on by the computer to place the stored value on the common input bus. Conversely, in output multiplexing, a number of d/a converters provide different voltage levels for the independent output channels. Each DAC is fed by a storage register, which holds its digital input word (and the corresponding analog output variable) until the computer feeds in the new, updating digital value.

Like track-holds, registers may be transparent, allowing input data to appear continuously at the output until a *strobe* signal causes the data to freeze; or, like sample-holds, they may be opaque, holding the last data byte until the strobe causes the data currently on the input lines to replace the existing output data.

More than one rank of registers may be used, to make the output data independent of changes of data at the input—an especially useful feature for d/a converters that must acquire from 9 to 16 bits of information from an 8-bit bus in two bytes. In the first rank, for example, for a 12-bit digital word, an 8-bit byte is acquired, and then a 4-bit byte; then their outputs are strobed into a 12-bit register to update the DAC simultaneously.

Some converters contain memory registers. Such devices have both analog (conversion) capability and multiple internal digital registers to permit independent update and readout.

Shift registers are used where data is transmitted serially over a single data channel (e.g., pair of wires) instead of as parallel bits over many wires. Data may be strobed in in parallel and out in serial, or arrive in serial and be strobed out in parallel.

DIGITAL DATA PROCESSORS

Converters are used with processors at all levels of system integration. While the distinctions are not clear-cut, and barriers are tumbling continually as technologies march ahead, it might be worthwhile to indicate a few general categories.

At the lowest level are the rudimentary converters found on digital IC chips designed for direct processing and/or communication of low-resolution analog signals.

Microprocessors will be found in games, small computers, instruments, display terminals, and data acquisition boxes and boards, usually designed for specific purposes (perhaps slaved to a nearby or distant host computer) and characterized by "intelligence."

Microcomputer and minicomputer systems may be designed for host-computer status in systems involving higher levels of capability—large amounts of memory, complex programs—including multitasking—number-crunching capacity, and interfacing flexibility. The converters they interface with may be in instruments, often using such standard media as RS-232 or IEEE-488 interfaces, or on analog input/output interface cards designed to plug into the system's bus configuration.

Mathematical processing of large amounts of data by computers can be speeded up, with a greatly reduced CPU burden, by the use of coprocessors or auxiliary specialized number-crunching modules, for example *array processors*, which perform such tasks as digital filtering, fast Fourier transformations (FFT), and convolution. At the heart of these operations are hard-wired digital multipliers and multiplier-accumulators, which can multiply (for example) two 32-bit floating-point numbers at 5-MHz and faster rates.

At the highest level are mainframe computers, which handle very large amounts of data, perform computations at very high speeds, and perform a great many independent tasks simultaneously. Their dealings with analog signals are likely to be at second hand by the downloading of instructions to lesser entities.

UP-DOWN COUNTERS

These devices, analogous to ramp generators, are quite useful for performing a variety of tricks with a/d and d/a converters. They are used in forming electronic servo loops for automatic error correcting, offset adjusting, long-term sample-holds, time-function generation with ROMs and DACs, etc.

In electronic servo applications, the up-down counter accumulates pulses representing the variable being controlled, adjusted, or measured, in much the same way that a servo-motor shaft accumulates rotational angle. The counter is often used in conjunction with a d/a converter to develop an analog value proportional to the accumulated count. The process is also used in tracking-type a/d converters and resolver/synchro-to-digital converters. Counters are also used for updating multiplexer channels sequentially in response to pulses.

FILTERS

Low-pass filters are used on the input side of an a/d converter to remove unwanted high-frequency components of the input signal. Noise and line-frequency interference can also be attenuated by filtering, but at the expense of

reduced response to fast input-signal amplitude variations. Filters (electrical or mechanical) are also used on the analog *output* from d/a converters, in order to smooth out the lumps created by discrete digital values. Filtering can be performed by digital techniques, using appropriate hardware and software, as well as by a wide range of active, passive, and sampling-type analog filters.

Pre-filtering is an important function, especially in high-speed information-processing systems, because the sampling frequency must be at least twice the highest frequency component of the signal input (Nyquist frequency). If higher frequencies are present, either within the signal or in the accompanying noise on the input channel, they can cause aliasing—intermodulation of unwanted high-frequency components of the signal (and input noise) with harmonics of the sampling frequency, to produce spurious signals at surprisingly low frequencies.

COMPARATORS

Analog comparators are important elements of data-acquisition systems. A high-gain analog comparator makes an elementary choice between the magnitudes of two inputs and decides which is the greater. This is the equivalent of a one-bit a/d conversion. Two comparators may form a decision "window". The a/d conversion process usually calls for a number of decisions; they may be made sequentially by a single comparator or simultaneously by a whole string of comparators, as in "flash" converters. Comparators may be free-running or latchable.

POWER SUPPLIES

Accuracy of interface systems is steadily rising, to the point where 12-bit resolution is quite routine, and 16-bit resolution is frequently needed for repeatability, resolution, linearity, and accuracy. Consequently, the design of the dc power system is no longer a trivial matter (it never really was!) since errors that remain third-order effects at 8-10 bits become menacing first-order effects at the 16-bit level. In many instances, careful separation of analog and digital grounds is required, demanding, in turn, considerable isolation between the various outputs that modern power supplies provide.

As with transducer selection, the priority of power-supply integration should be raised from the status of an afterthought. Too often, the power supply design (or choice) is left until last, where it is presumed to be able to take up all the slack or tolerances that other design stages create. Instead, power supplies (ac/dc or dc/dc) deserve at least as much initial attention at a competent engineering level as the selection of converters, amplifiers, sample-holds, multiplexers, and other devices.

A related question is whether to use power supplies and/or regulators in large, medium, or small chunks, for major portions of the system, for individual chassis, or perhaps even for mounting on individual boards. The issues in-

volve space, cost, circuit independence vs. excessive lead length and wire size, connector technology, avoidance of ground loops, allowable local dissipation levels, etc., and must also include the question of interruptibility.

If continued operation despite loss of primary power is essential, either from considerations of overall system reliability or because of potential loss of data in volatile memory, the system design should include a determination of what level of operation or memory retention is necessary in case of an interruption—and some arrangement for continuity of supply, for detection when standby power is in use, and for contingency decisions or alarm.

DIGITAL PANEL INSTRUMENTS

These devices form a kind of self-contained digital processing system all by themselves, often comprising not only conversion and display, but also multiple-channel scanned inputs and signal conditioning (an example is the Analog Devices AD2036 6-Channel Scanning Thermocouple Thermometer). They can be either "dumb" or intelligent and can be used in conjunction with other digital equipment, since most DPIs have digital outputs available.

At its core, a DPI may be regarded as an (often self-powered) independent analog-to-digital converter—often including signal conditioning—complete with case, overrange capability, input protection, visual readout, and digital output, usually in parallel or multiplexed BCD (binary-coded decimal) format. Thus, the DPI can be used as a stand-alone converter that interfaces with both humans and machines.

DEDICATION

As we have already noted, data systems exist to serve human purposes. It is to all these humans, who design, program, use, play, and work (or are studying to do so) with systems that employ converted analog data, that this book is fraternally dedicated.

Chapter Two

Data Acquisition

Analog data is acquired in digital form for any or all of the following destinations:

Storage	Processing
Transmission	Display

Digital data may be *stored* in either raw or processed form; it may be retained for short, medium, or long periods. It may be *transmitted* over long distances (for example, to or from outer space), or short distances (from one part to another of a microprocessor-based instrument). The data may be printed on a printer or plotted on an X-Y plotter for a permanent hard copy, or it may be *displayed* on a digital panel meter, as part of a cathode-ray-tube presentation, as speech or other sounds, or in any other form that stimulates human senses.

Processing can run the gamut from simple comparisons to complicated mathematical manipulations. One might use it for such purposes as collecting information, converting data to a useful form, using the data for controlling a physical process, performing repeated calculations to dig out signals buried in noise, generating information for displays, simplifying the jobs of warehouse employees, controlling the color of paint, the thickness of a wrapper, maneuvers in a game, the speed of a subway train.

But it all starts with getting the data in digital form, as rapidly, as frequently, as accurately, as completely, and as cheaply as necessary.

The basic instrumentality for accomplishing this is the analog-to-digital (a/d) converter (ADC). It can be an IC chip, a shaft digitizer, a DPM with digital outputs, or a sophisticated high-resolution high-speed device; physically, it

may take the form of a box, a card, a potted module, or an integrated circuit. It may be functionally integrated with other elements.

To accommodate the input voltage to the specified conversion relationship, some form of scaling and offsetting (signal conditioning) may be necessary, performed with an amplifier/attenuator. To convert analog information from more than one source, either additional converters or a multiplexer may be necessary. To increase the rate at which information may be accurately converted, a sample-hold may be desirable. To compress an extra-wide analog dynamic range, a logarithmic amplifier or conversion relationship may be found useful.

The properties which the data-acquisition system must have depend on both the properties of the analog data itself and what is to be done with it. This chapter deals with aspects of signal flow, from sensors through conversion. Chapters 4 and 6 have to do with integration into systems and instruments, Chapter 7 deals with the fundamentals of ADCs, Chapter 8 deals with some of the forms IC ADCs can take, and the Chapters in Part III deal with ADCs designed for special purposes, such as high-speed ("video") applications. A great deal more information on the nature and handling of analog signals from transducers in preparation for digital data acquisition can be found in the *Transducer Interfacing Handbook*[1].

In this chapter, we shall introduce some of the functional architectures that have proven useful and popular and discuss some of the considerations involved in the choice of configuration, components, and other elements of the system. Additional information will be found throughout the book.

2.1 THEN AND NOW

A quarter-century ago, a/d converters capable of 0.05% performance and 50,000-sample-per-second conversion rates cost about $8000, consumed about 500 watts, and occupied about one-quarter of a cubic meter.

Today, the completely self-contained monolithic Analog Devices AD573 requires less than 20 microseconds for a 10-bit 0.05% conversion, is available in quantity at less than 0.2% of the price, and is packaged in a 20-pin plastic DIP. And it is designed for easy interfacing with the modern microprocessor.

The space formerly occupied by the converter alone will now hold (for a similar order of dollar investment) a MACSYM 150: a complete multitasking minicomputer-based data-acquisition system including computer, keyboard, display, disk drive, converter, and a set of input-output cards.

In the past 25 years, as the above examples show, the processing power and complexity of data-acquisition and computer hardware have increased radically, thanks primarily to the semiconductor revolution. Sophisticated software and interactive terminals, which make computer techniques accessible

[1]Sheingold, ed., *Transducer Interfacing Handbook* (Norwood MA 02062: Analog Devices, Inc., 1980)

even to small children, permit interconnections, switching, and "knob twisting" to be performed under remote or programmable control. All of this, available at prices that were then undreamed of, has brought matters to the point that digital, rather than analog, "massaging" of information is a matter of routine, rather than exotic necessity.

What have not changed, however, are the fundamental system problems confronting every digital data-system designer. Of course, it helps to have small, quiet, low-cost, cool, low-current-drain components. But the designer is still up against the laws of Mother Nature, who often prefers to keep her secrets safely obscured by noise, EMI, ground loops, power-line pickup, and transients induced in signal lines from machinery. Separating the signals from these obscuring effects, then, becomes a challenge to ingenuity and imagination, coupled with a great deal of experience and persistence. Design is not merely a matter of purchasing fast, high-resolution a/d converters—but having them available at realistic cost is an incentive for giving them useful jobs.

2.2 ENVIRONMENT AND COMPLEXITY

Though there are many ways of starting to think about data-acquisition systems, a highly relevant approach has to do with environment. Some systems are intended to operate with modest accuracy in hostile environments (factories, vehicles, military surroundings, and remote installations); others are best suited to making high-precision measurements in such (probably fictitious) favorable surroundings as electrically quiet laboratories at room temperature; and yet others may be mixed, acquiring analog data generated in hot, noisy environments, processing it in the security of a quiet control room, and transmitting the resulting digital data through a world rife with interference.

Environment may give rise to such considerations as:

Analog vs. digital signal transmission
Signal accuracy vs. waveform recovery
Isolation vs. direct wiring
Subminiature vs. macroscopic size
Simple vs. complex system architectures
Integrated vs. distributed approaches
Local vs. remote processing
"Hi-rel" vs. "commercial" parts
Choice of power supplies and physical hardware

Hostile environments manifest themselves in combinations of physical, chemical, and electrical challenges. *Physically* hostile environments may exhibit extremes of temperature, pressure, acceleration, humidity, and radiation (both ionizing and non-ionizing). *Chemically* hostile environments may involve such corrosive surroundings as salt spray, biological fluids, noxious atmospheres, dirt, and chemically active fluids and gases. *Electrically* hostile

environments may include destructively high voltages and magnetic fields, as well as dc, ac, and transient interference over the whole spectrum.

In addition to these actively hostile forms of environment are "passively" hostile environments that must be disturbed as little as possible—physically, chemically, or electrically—by the introduction of data-acquisition equipment. Some environments are both actively and passively hostile.

Typical actively hostile environments might include the wide temperature range of aircraft engines, shock and vibration in railroad freight cars, moisture and corrosiveness in oceangoing equipment, high radiation levels in the vicinity of nuclear reactors, and combinations of many of these factors in geothermal wells. In passively hostile environments, the data-acquisition equipment (or portions of it) may be required to fit into small physical space, not raise the local ambient temperature substantially, not generate excessive electromagnetic interference, not be made of materials that would interact chemically with sensitive surroundings, and not use more than a small part of the power furnished to or generated within its surroundings.

On the other hand, for laboratory-instrument applications, the system designer's problems may be related more to the performing of sensitive measurements (usually under favorable conditions, with respect to electrical interference) than to the gross problems of protecting either equipment or the integrity of analog data. *However, with the increasing use of microprocessors and the promiscuous use of both analog and digital circuits in precision instruments, freedom from electrical interference cannot ever be taken for granted.*

Systems existing in hostile environments may require electronic devices capable of wide-temperature-range operation, excellent shielding, considerable design effort aimed at eliminating common-mode errors and preserving resolution, early conversion and digital data-transmission (perhaps via optical fibers), redundant paths for critical measurements, and—in some instances—considerable processing of the digital data to extract the maximum of information.

Measurements in the laboratory, with narrower temperature ranges and fewer sources of ambient electrical interference, may be easier to make and communicate, but higher accuracies (or resolutions) may require more sensitive devices, plus a still-considerable degree of effort to preserve appropriate signal-to-noise ratios.

2.3 KEY FACTORS

Environmental factors aside, the choice of configuration and circuit building blocks in data acquisition depends on a number of critical considerations, among them:

Resolution and accuracy
Number of analog channels to be monitored
Sampling rate per channel

Throughput rate
Signal-conditioning requirements
Intended disposition of converted data
The cost function

Besides the choice of appropriate component performance levels, careful analysis of the above factors is required to obtain the lowest-cost circuit configuration to obtain the desired overall performance.

Commercially available data-acquisition systems range from basic a/d converters to multichannel converters on monolithic chips to completely integrated systems on cards and in boxes, and they even include converters that are functionally inseparable from the digital processor. In later chapters, we will consider the various optional levels of integration available in a single package or piece of equipment. However, in this chapter, we will treat the functions peripheral to the a/d converter *as though they were embodied by separate components,* in order to make clear the architectural choices that a designer might have, with their characteristics, advantages, and disadvantages.

Typical configurations include:

Single-Channel possibilities
- Direct conversion
- Sample-hold and conversion
- Preamplification
- Signal-conditioning

Multi-channel possibilities
- Multiplexing the outputs of single-channel converters
- Multiplexing converter inputs
- Multiplexing the outputs of sample-holds
- Multiplexing the inputs of sample-holds
- Multiplexing low-level data
- More than one tier of multiplexers

Some of the more-interesting signal-conditioning options include:

Ratiometric conversion

Wide-dynamic-range options
- High-resolution conversion
- Range biasing
- Automatic gain switching
- Logarithmic compression
- Logarithmic conversion
- Digital correction of analog errors

Noise-reduction options
- Filtering
- Integrating-type converters
- Digital processing

Finally, in evaluating tradeoffs, there are at least three types of "budgets" that should be considered: error budget, system time budget, and the relationship of the "make-or-buy" question to the cost budget at a given level of integration.

2.4 SINGLE-CHANNEL CONVERSION SUBSYSTEMS

Direct Conversion

Figure 2.1 represents the simplest digitizing system, a lone a/d converter performing free-running repetitive conversions at a rate determined by the time for a complete conversion. It has power inputs and an analog signal input. Its outputs are a digital code word—which may include overrange indication—in parallel, byte-serial, or serial form; polarity information (if the analog input is bipolar); and a "status" output that indicates when the output digits have become valid.

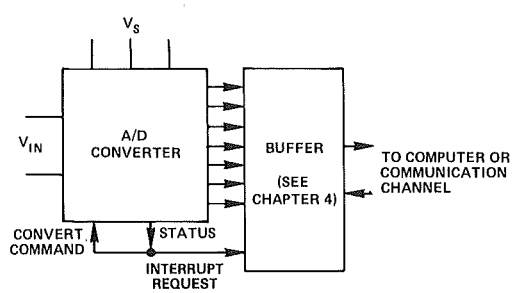


Figure 2.1. Single-channel free-running a/d converter.

Perhaps the best-known converter of this kind is the basic digital panel meter circuit, which consists of a basic a/d converter and a numerical display. For many applications, the sole purpose of digitizing is to obtain the display of the digits, i.e., to house the DPM circuit in a box and use it as a meter rather than as a system component. The DPM, however, is not necessarily the best way to digitize a single channel. Its two major shortcomings are: it is slow, and its BCD digital coding must be changed to binary if its output is to be processed by binary equipment. When free-running with a system, its output is strobed in following an Interrupt when the data becomes valid, rather than by a system interface command.

Converters designed for system applications (including many DPMs) can usually receive external commands to convert or hold. For dc and low-frequency signals, the converter is usually a dual-slope type (see Chapter 7), which has the advantage that it is inherently a low-pass filter, capable of averaging out high-frequency noise and nulling frequencies harmonically related to its integrating period. (For this reason, the integrating period is usually made equal to the period of the line frequency, since the major portion of system interference usually occurs at that frequency and its harmonics.)

The "actual" value of input that is converted by an integrating-type converter is represented by the average over the signal-integration interval. Since that interval is a fraction (about one-third, but not necessarily constant) of the total time required for the conversion cycle, what one can say about the value of the signal and when it occurred is that the digital output represents the most probable value during a significant portion of the conversion period.

For dc voltages or signals known to have constant values during the conversion interval, even if the changes occur rapidly (as long as they have settled prior to the start of a conversion), any new value will be converted to specified resolution and accuracy within the conversion interval.

For repetitive conversions of continuously changing inputs, however, the maximum rate at which the input signal can vary and still permit the converter to resolve 1 least-significant bit (LSB) of binary output, irrespective of the waveform, is

$$dV/dt = 2^{-n} V_{FS}/T \tag{2.1}$$

where V is the input, n is the number of bits of binary resolution, V_{FS} is the full-scale span, and T is the time between conversions. The maximum rate of change is thus 1 LSB per conversion period.

If $V = (V_{FS}/2) \sin 2\pi ft$, then

$$dV/dt = (V_{FS}/2)\, 2\pi f \cos 2\pi ft \tag{2.2}$$

and dV/dt_{max} is equal to the magnitude, $(V_{FS}/2)\, 2\pi f$. Thus,

$$2^{-n}/T = \pi f \tag{2.3}$$

and the maximum sine-wave frequency that can be converted with 1-LSB resolution is

$$f = 2^{-n}/(T\pi) \tag{2.4}$$

For example, if the conversion rate for a 12-bit binary integrating-type converter is 25 per second (T = 0.04s), f = 0.002Hz, corresponding to 0.12V/s of a 20-V span. For faster dV/dt's, changes cannot be resolved to within 1LSB during a conversion period.

So far, the context has been that of the dual-slope integrating a/d converter, which spends about 1/3 of its sampling period performing an integration, and the remainder of the time counting out the average-value-over-the-integrating-period as a digital number, and resetting to initial conditions for the next sample. Though slow, the integrating a/d converter can be readily manufactured in integrated-circuit form and is quite useful for measurements of tem-

perature, battery discharge, and other slowly varying voltages, especially in the presence of noise.

However, by far the most popular type of converter for system work is the successive-approximation device (Chapter 7), since it is manufacturable as an integrated circuit and is capable of high resolution (e.g., 16 bits), high speed (e.g., 1μs for 12-bit conversion), and quite reasonable cost.

The successive-approximation converter, used by itself, has the weakness that, at higher rates of change, it generates substantial linearity errors because it cannot tolerate change during the weighing process. The converted value will be at some value between the extreme values occurring during conversion, and the time uncertainty approaches the magnitude of the conversion interval. Figure 2.2 illustrates this point. Finally, even if the signal is varying slowly enough, noise rates-of-change (perhaps introduced by the signal itself) that are excessively large will cause erroneous readings that cannot be averaged, by either analog or digital means.

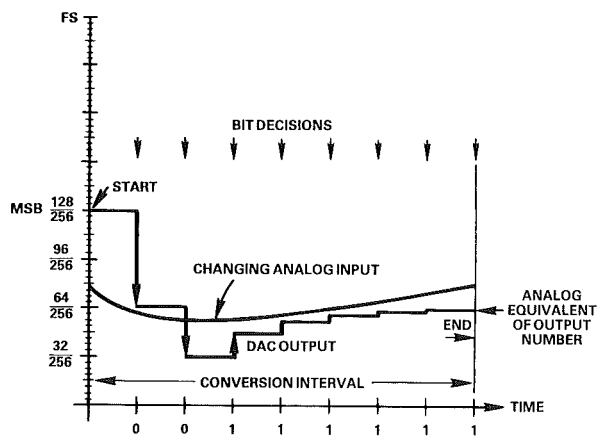


Figure 2.2. Error of successive-approximation 8-bit converter with changing input during conversion. The input is at the same level at the beginning and end of conversion, but the output value is about 16 LSBs low.

Since the final converted value occurs at an unknown time during the conversion interval, the time uncertainty corresponds to the conversion interval. In the example of Figure 2.2, a value equivalent to the output number occurs twice, at about 1/8 and 5/8 of the conversion interval. For a sinusoidally varying input, the relationships expressed in equations 2.1 through 2.4 apply. In the above example, if T, for a successive-approximation converter, is 1.5μs for 12 bits, the maximum allowable frequency for maintaining bit-at-a-time resolution of a sine wave becomes 52Hz, and the maximum rate of change for 1-LSB resolution of a 20-V span becomes about 1,600V/s—an improvement over the integrating converter, but far from sensational.

Sample-Hold and Conversion

A converter can be made to operate at considerably greater accuracies at high speeds with precise timing of samples, irrespective of the time required to complete a conversion—overcoming the weaknesses mentioned above—by introducing a *sample-hold* (or *track-hold*) between the input signal and the converter's input (Figure 2.3).* Between conversions, the device may acquire and track the input signal. Just before a conversion is to take place, it is switched to *hold* and remains in that state throughout the conversion.

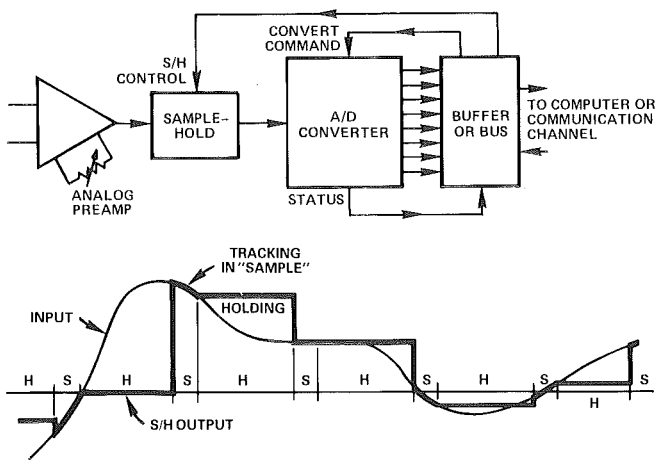


Figure 2.3. Sample-hold in single-channel data acquisition—block diagram and waveforms.

It can be seen that, if the S/H responds instantaneously and accurately, the converter can accurately convert signals having rates of change of any magnitude, at sampling rates up to the ADC's maximum conversion rate. In practical sample-holds, however (Chapter 18), there will be such time-related errors as acquisition time, tracking delay, and aperture time. Typical values of these parameters for track-holds designed to be used with fast a/d converters are 5μs acquisition time to 0.01%, 50ns tracking delay, and 25ns aperture time, with 0.5ns uncertainty (jitter). If the acquisition time is adequate, and aperture time and tracking delay: compensate one another, can be nulled by phase adjustment if necessary, or are unimportant in repetitive sampling as long as they are consistent, the principal (irreducible) source of time error in sampled-data systems is the *aperture uncertainty*.

When a sample/track-hold is used with an a/d converter, the signal is frozen as of the instant *hold* is achieved; thus, T in equations (2.1) through (2.4)—the uncertainty as to when the signal was sampled—represents the aperture uncertainty (instead of the much longer conversion time). If a track-hold with

*Strictly speaking, a *track-hold* (T/H) tracks the input until the sampling ("hold") command is received, then holds; a *sample-hold* (S/H) remains in hold until commanded to get a sample ("sample"), then quickly returns to hold. In most cases, the same device can be used in either fashion.

0.5ns aperture uncertainty is used with the 12-bit, 1.5μs converter mentioned earlier, the maximum-frequency sinusoidal signal that can be converted with 1-LSB resolution is $(2^{-12})/(0.5 \cdot 10^{-9} \cdot \pi) = 155\text{kHz}$ (from 2.4), corresponding to a maximum rate-of-change of about 9.8V/μs. The sample-hold should have enough bandwidth to deal accurately with the signal amplitude (for stationary ensembles)—but also with phase for unique events or where phase between two channels is critical.

Figure 2.4 shows that—in contrast with Figure 2.2—at the end of each conversion interval, the converter, with a constant input applied by the S/H, will deliver an accurate digital representation of the input value as it was at the start of conversion. If the converter is accurate, any errors that are functions of time will be due to errors of the sample-hold, including the acquisition errors mentioned above, plus droop during the conversion interval, and any linearity, offset, and transient errors. Band-limited random noise present on regularly sampled signals, before sampling and conversion, may be susceptible to digital averaging by the processor.

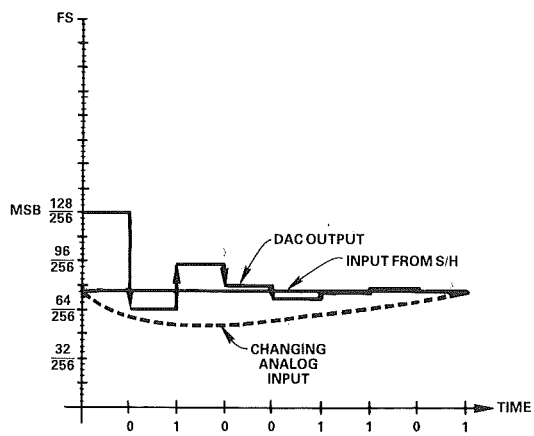


Figure 2.4. Same converter and input signal as in Figure 2.2, but with converter preceded by sample-hold, the changing input signal is ignored. Converter produces the correct output as of the instant of *hold*.

Sample, Signal, Noise, and Aliasing In order to avoid errors due to an insufficient number of samples, the Sampling Theorem tells us that regularly spaced sampling must occur at least at the *Nyquist rate*, twice the frequency of the highest-frequency signal or noise component; that is, either a sufficiently high sampling rate must be employed or else all components of signals and noise at frequencies equal to or greater than the *Nyquist frequency*, i.e., one-half the sampling rate, must be filtered out before sampling. Since practical filters require a compromise between attenuation in the pass band and transmission in the stop band, the sampling rate is often three or more times the filter cutoff frequency.

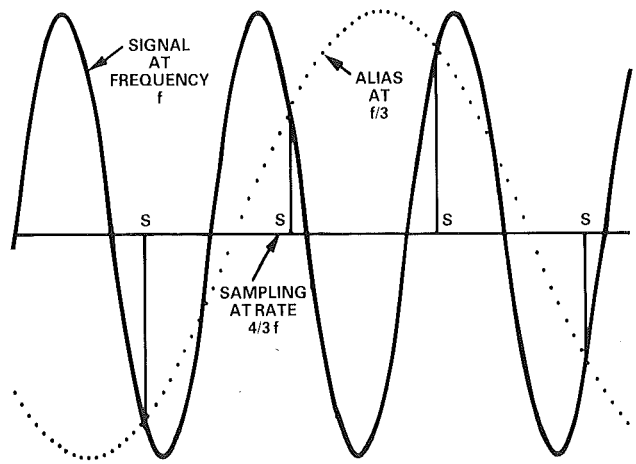


Figure 2.5. Example of aliasing. A sinusoid is sampled (S) at 4/3 of its frequency (f). The resulting set of samples form an alias at (4/3 – 3/3)f = (1/3)f, i.e., a train of pulses with amplitude and timing indistinguishable from those associated with a sinusoid at f/3.

If analog signals at higher frequencies are present, the sampling process will produce sum and difference frequencies with the sampling frequency and its harmonics; the difference frequencies, in particular, will produce spurious low-frequency signals, or *aliases*—in the signal passband—that cannot be distinguished from the signal. Figure 2.5 is a simple example illustrating aliasing.

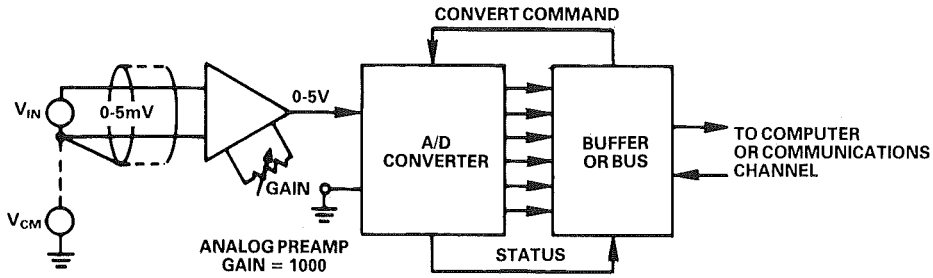
Since sample-holds usually operate at unity gain*, with errors referred to full scale (which should be the same as the converter's full-scale range), scaling or preamplification should usually occur before the signal is applied to the sample-hold.

Preamplification (Figure 2.6)

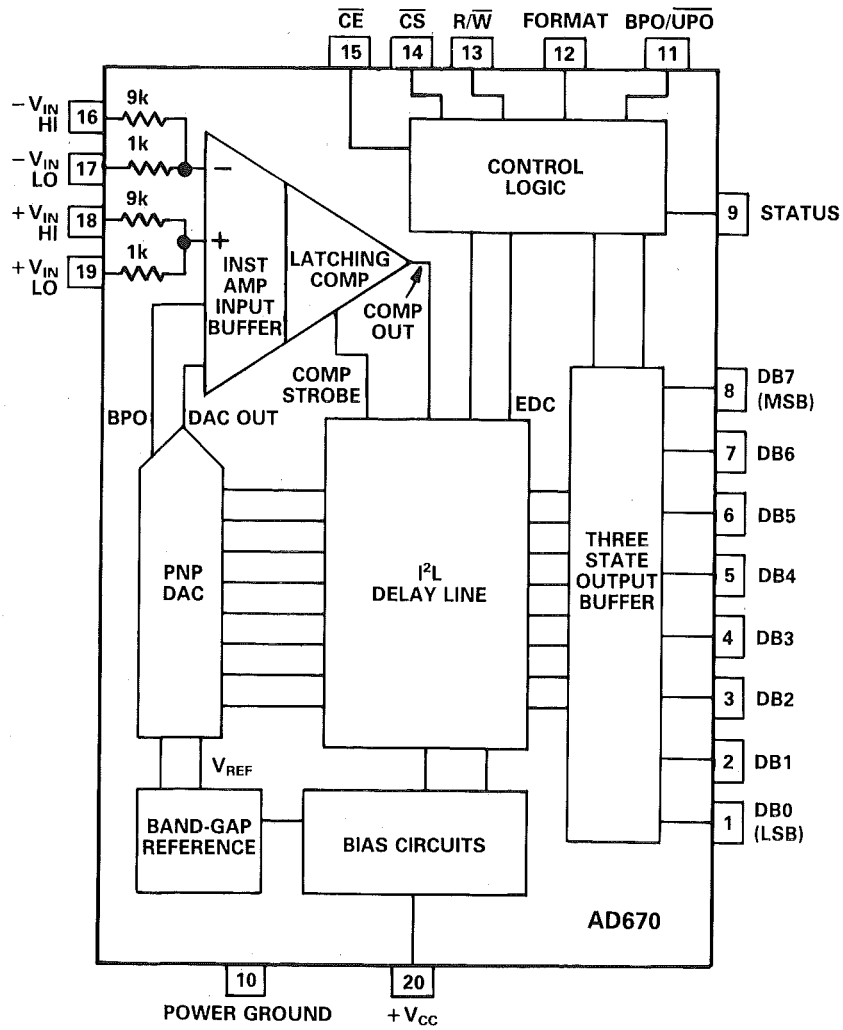
In most cases, converters designed as components are "single ended" with respect to power common† and have normalized input ranges of the order of 5 or 10 volts, single-ended or bipolar. It makes sense to scale signal inputs up or down to the standard converter input level, to make fullest possible use of the converter's available resolution.

Figure 2.6a shows a typical preamplifier configuration. The preamplifier should have low dynamic output impedance, because the inputs of some types of a/d converters may have large current pulses, which will load the preamp's output and can cause errors. Sometimes these functions are combined with the converter. For example, in Figure 2.6b, we see a block diagram of

*Some types, however, do permit adjustment of gain and input configuration by a choice of external circuitry.
†Some converter designs and digital panel meters, have differential, and even isolated, floating inputs, and provide signal gain.



a. A/D converter with preamplifier/signal conditioner.



b. Block diagram of an 8-bit ADC with on-chip signal conditioning.

Figure 2.6. A/D Converter and preamplifier.

monolithic 8-bit a/d converter with a built-in gain-of-10 differential preamplifier and jumper-programmable choice of gains and input configuration (see Section 8.6.1). The on-chip preamplifier also buffers the input source from the conversion process.

Single-channel data-acquisition systems are available that include a programmable-gain amplifier (PGA)—with gain controlled by switching of resistors, either manually or under software control, plus a sample-hold function and an a/d converter in a single modular or integrated-circuit package.

If the signals are of reasonable magnitude (already preamplified), and already exist within a system referenced to a good-quality common "ground," the scaling may be simply accomplished with operational amplifiers in a single-ended or differential configuration. As is more generally the case, if the signals are from outside the system (or subsystem, or board, or neighborhood), or are quite small, or have an appreciable common-mode component, a differential instrumentation amplifier may be profitably used, with characteristics that depend on the gain required, the signal level, the needed CMR, bandwidth, impedance levels, and cost tradeoffs.

If the input signals must be galvanically isolated from the system, an isolation amplifier must be used to break all conductive signal paths. Such amplifiers generally employ optical or magnetic coupling. Isolation is mandated for protection of patients and subjects in clinical medical-instrument applications; it is also useful where common-mode spikes are encountered, as well as for industrial applications requiring *intrinsic safety* and for applications in which the signal source is at a high off-ground potential.

Signal Conditioning

This blanket term includes a wide variety of analog-to-analog possibilities. Many of the functions could also be performed digitally, under program control, depending on availability of processing capability, tradeoffs of cost, speed, accuracy, hardware vs. software, level of system integration, and the designer's personal orientation.

Signal-conditioning devices are available in a variety of packages and capabilities to meet the needs of the system designer. For example, instrumentation, isolation, and thermocouple amplifiers are available in IC packages, signal conditioners are available as board- or track-mountable modules, and whole families of modular signal conditioners for different purposes are available for mix-match mounting in expandable manifolds that include power supplies.

Here are some instances of signal conditioning. Scaling of input gains to match the input signal to the converter's full-scale span, using op amps or an instrumentation amplifier, is a simple, obvious example. One might also include dc offsets to bias odd ranges (for example 2-to-10 volts, derived from a 4-to-20mA instrumentation current loop via a 500-ohm load resistor) to

levels more compatible with standard converters. Preamplification has already been mentioned.

Linearizing of data from thermocouples and bridges can be performed by analog techniques, using either piecewise-linear approximations (generated by biased-diode circuitry) or smooth series-approximations, using low-cost IC analog multipliers. It can also be done digitally, after conversion, by performing the necessary calculations with a microprocessor or by storing the inverse or complementary function in a read-only-memory (ROM) lookup table.

Analog differentiation can be used to measure continuously the rate at which the input varies; integration could be used to obtain total dosage from a rate of flow. Either could be used to produce a 90-degree phase shift; an op amp, connected as a simple all-pass filter, can be used to provide an arbitrary phase shift. Sums and differences could be used to reduce the number of data inputs (analog data reduction).

Analog multipliers canbe used to compute power by squaring voltage or current signals, or multiplying them together. RMS-to-dc converters compute rms directly. Analog dividers of various types could be used to compute ratios or the logarithms of ratios, or square roots. Devices that compute $Y(Z/X)^m$ can take ratios over wide dynamic ranges and perform analog function fitting.

Comparators can be used to make decisions based on analog levels (e.g., to convert only when an input exceeds a threshold or is within a "window"). Op amps and diodes may be used to perform simple "ideal-diode" functions.

And—what seems like getting "something for nothing"—logarithmic circuits can be used for range compression to permit the conversion of signals having wide dynamic ranges with converters having considerably less resolution than would be otherwise required.

Active filters are essential elements to minimize the effects of noise, carrier frequencies, and unwanted high-frequency components of the input signal. The increasing interest in filtering is reflected in the growing number of books and availability of hardware and software devoted to both analog and digital filter design. Analog filters can be either fixed or digitally programmable, using d/a converters.

One could go on and on but the basic point should have been made: that in system design, all data-processing need not be digital. Analog circuits can perform remote or local processing or data reduction effectively, reliably, and economically, and should be considered as alternative ways of reducing software complexity, noise, board space, and—quite often—cost. Figures 2.7 through 2.10 show a few examples.[2]

[2] For many more examples, see Sheingold, ed., *Nonlinear Circuits Handbook* (Norwood MA 02062: Analog Devices, Inc., 1974). Many additional examples can also be found in (1).

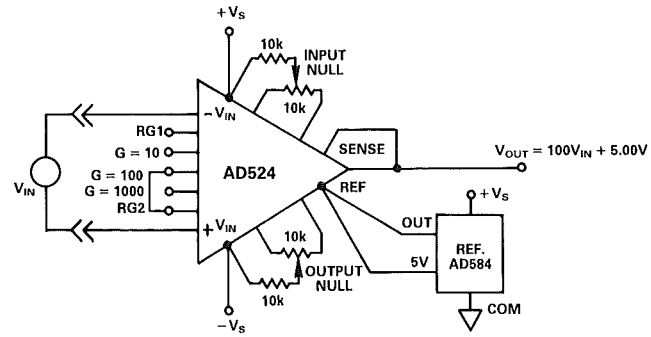


Figure 2.7. Instrumentation amplifier provides offset and scaling. Here, the gain is 100V/V, and precision offset is +5.00 V.

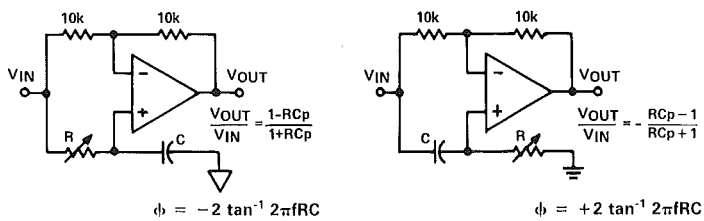


Figure 2.8. Operational amplifier generates an arbitrary phase shift. Gain = +1, all pass.

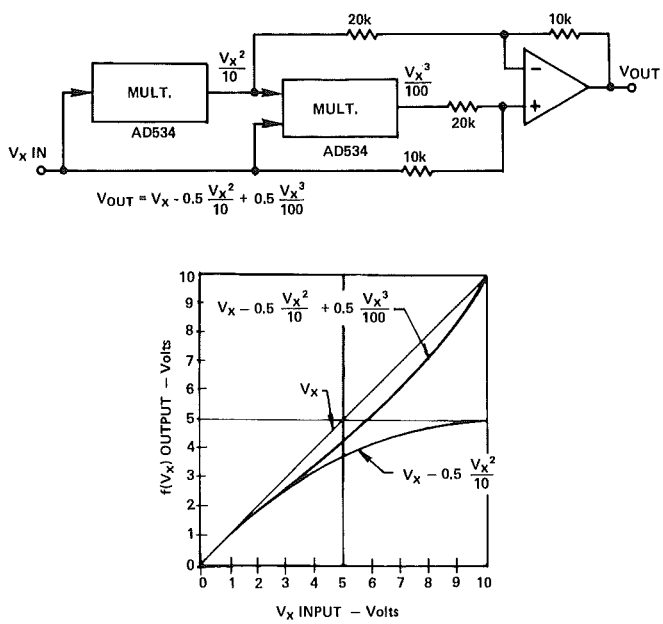


Figure 2.9. Using multipliers for nonlinear functional relationships, such as linearization.

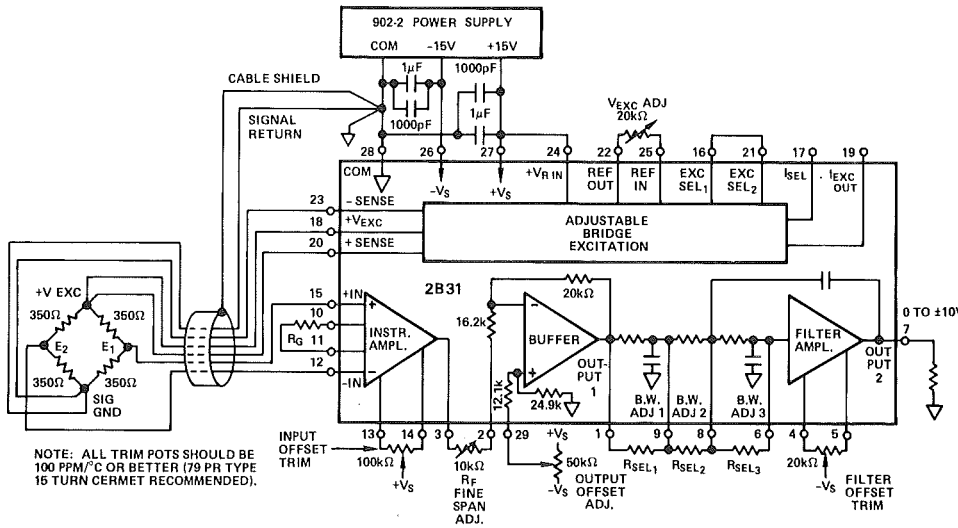


Figure 2.10. Using a signal-conditioning module for bridge excitation, preamplification, and filtering.

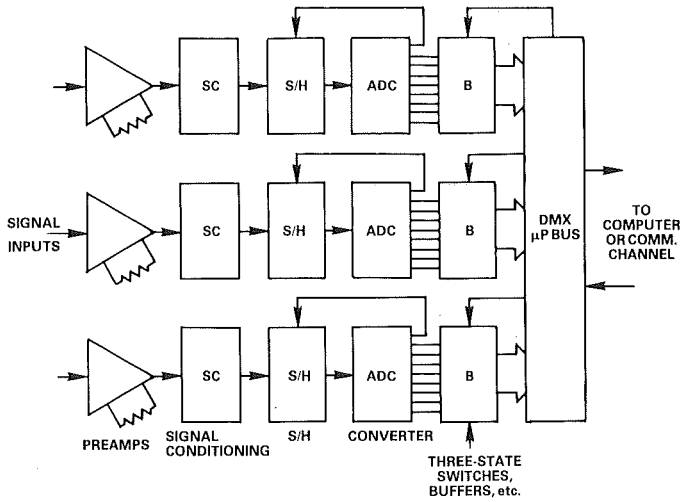
2.5 MULTI-CHANNEL CONVERSION

In multi-channel systems, elements of the acquisition chain may be shared by two or more input sources. This sharing may occur in various ways, depending on the desired properties of the multiplexed system. Large systems may combine several different kinds of multiplexing, as well as cascaded tiers of the same kind. Chapter 4 has more information about multiplexed data acquisition in practice at higher levels of integration, and Chapter 19 has further information about multiplexers and switches.

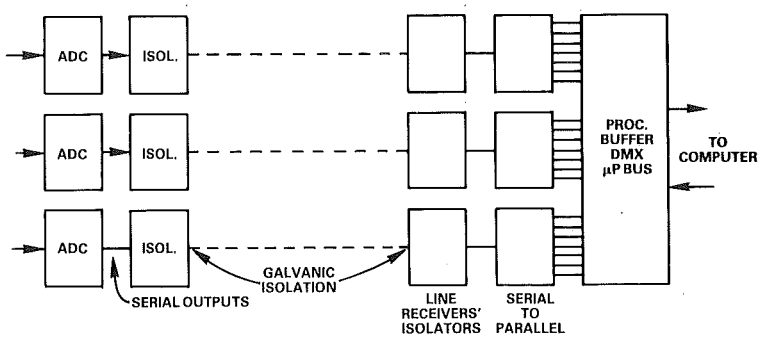
Multiplexing the Outputs of Single-Channel Converters

Although the conventional way to digitize data from many analog channels is to introduce the time-sharing process at the analog portion of the system by multiplexing the input of a single a/d converter among the various analog sources, in sequence, an alternative parallel conversion process is becoming increasingly practicable. Cost of a/d converters has dropped radically in recent years, and it is now possible to assemble a multi-channel conversion system, with the seeming extravagance of one converter for every analog source, yet considerably improved performance at reasonable cost (Figure 2.11).

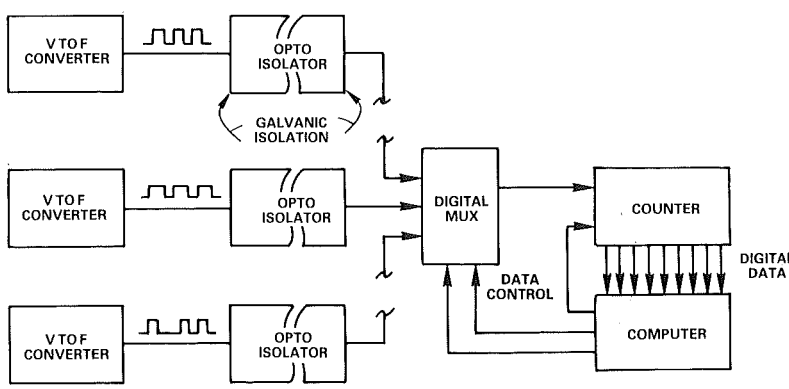
This parallel conversion approach has its advantages. First, a desired overall digital throughput rate can be had with slower converters; alternatively, the converter-per-channel may run at top speed, providing a much greater flow of data into the digital interface. For a modest data rate, however, with more channels (and fewer conversions per channel), it may be possible for sample-holds to be eliminated, at a cost saving. Fewer conversions might also mean that a slower converter could be used, generally resulting in even further cost savings, especially since some channels may not require high resolution.



a. Basic multi-channel conversion scheme, using digital multiplexing.



b. Multi-channel conversion using remote a/d converters.



c. Multi-channel conversion using v/f converters and digital multiplexing.

Figure 2.11. Multi-channel conversion schemes.

The bus structures used by systems employing microprocessors encourage the use of digital multiplexing, with all devices connected to the bus via 3-state switches, enabled selectively by "chip select" logic signals from decoders and read/write control signals (*write* to initiate conversions, *read* to obtain the results). The converter's status line can provide *interrupt* signals indicating "conversion complete—data is ready".

The parallel-conversion approach provides a further advantage when applied to industrial data-acquisition systems, where strain gages, thermocouples, thermistors, etc., are strung out over a large geographical area. In essence, by digitizing the analog signals right at their source and transmitting serial digital data, rather than the original low-level analog signals (Figure 2.11b), a considerable immunity to line-frequency (50-60-400Hz) pickup and ground-loop interference is achieved. Among other factors, the digital signals can be coupled optically or even transmitted via fiber-optic links for complete electrical isolation and total indifference to electrical interference.

A multichannel array of voltage-to-frequency (v/f) converters is an interesting way of transmitting data generated by slowly varying signals, with dynamic ranges of up to 10^6—and requiring accuracies to within 0.01%. The outputs are TTL pulse trains, which may be easily isolated optically. The output of each VFC, in turn, is counted and read. As Figure 2.11c shows, the computer controls the multiplexer and acts as a time base for the counter.

Not least, among the subtle benefits of digitizing sensor signals at their source, is the ability to perform logical operations on the digitized data before it is fed into the computer. In this way, for example, mainframe involvement with data is streamlined and redundancies are minimized. More specifically, remote processing makes it possible, for example, to access data from slowly varying thermocouple sensors less frequently, while reading in data from rapidly changing critical sources at enhanced speed. In fact, the versatility of a digital subsystem may be exploited to make its own decision as to when a particular data channel should be brought to the attention of the computer by means of Interrupts. If certain signal sources remain constant or within a narrow range for long periods, then change rapidly later in the process, it is possible to ignore these data until the changes occur. (A local microcomputer can store the stationary values and make the decisions.)

In sum, a great deal of flexibility and versatility is gained by changing the interface process from analog multiplexing to digital multiplexing. Logic decision circuits or local microprocessors can exercise judgement on when and what data to feed the host computer and, in general, can give the overall interface a much larger measure of autonomy than is possible with an entirely analog conversion system. (Systems involving analog multiplexing have a Catch 22: The computer cannot make decisions about the data submitted by an analog multiplexing system until it has received the data upon which to base its judgements... this means that the data have been converted and interfaced

before the computer can decide that a particular piece of information is redundant. And there is no guarantee that it will be redundant on the next pass.)

Finally it should be noted that if, for example, the data is being transmitted from a lunar vehicle to Earth, the channel is quite crowded, and the sort of *redundancy-reduction data compression* described above is absolutely essential to make sure that the items of data that get through are those having the highest priorities, by virtue of containing intelligence rather than redundant information.[3]

For each channel of the digitally multiplexed system, there could be the chain described earlier: preamplifier, signal conditioning, sample-hold, converter. It is also possible that, for one or more of the channels, there are a number of multiplexed subchannels, especially if they are carrying similar information.

Examples of Multiplexed Converter Inputs

In some scanning-type panel meters (such as the AD2037), the input channels undergo multiplexing, followed by signal conditioning and a conversion, in which the analog input circuitry is isolated from the digital logic circuits.

An unusual single-chip integrated-circuit data-acquisition system, the 8-bit AD7581 (Figure 2.12), multiplexes and converts 8 channels of analog data in sequence continuously, and stores the most recent value of each channel in a separate memory register, where it may be addressed and accessed via a microprocessor data bus at any time.

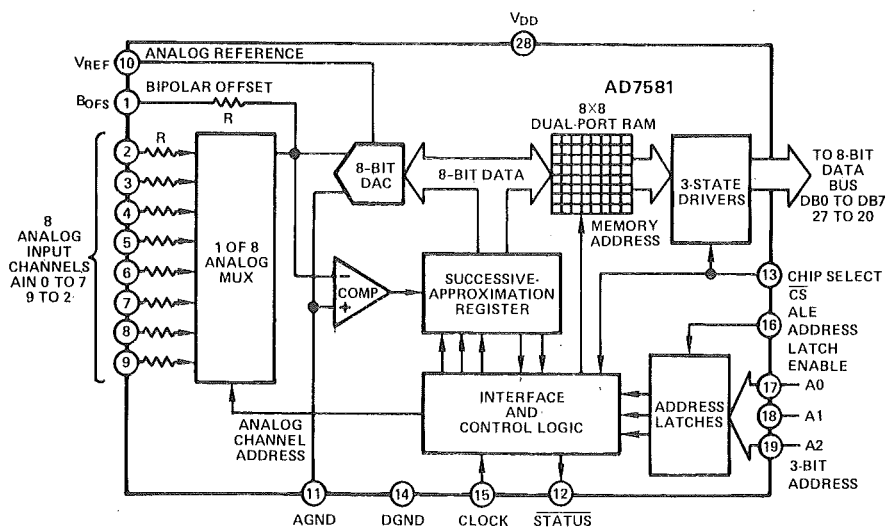


Figure 2.12. Block diagram of 8-channel 8-bit memory ADC.

[3] "New approaches to Data-Acquisition System Design," by T. O. Anderson, *Analog Dialogue* 5-1, 1971

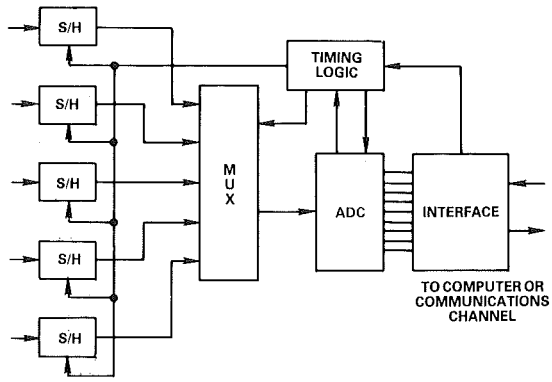


Figure 2.13. Multiplexed system with simultaneous sampling, sequenced conversion.

Multiplexing the Outputs of Sample-Holds

Working back from the interface (with a minimum number of shared elements, except for the bus) towards the more conventional situation, in which the number of shared elements is maximized, we consider the intermediate case of a shared a/d converter, with a multiplexer at its input, switching among the outputs of a number of sample-holds (Figure 2.13). This configuration is found where sample-holds are updated rapidly, perhaps even simultaneously, or at critical instants for individual inputs, then read out in some sequence. It is generally a high-speed system, in which all items of data delineating the state of the system must do so for the same given instant. Multiplexing may be done sequentially or, when required, by random addressing. The sample-holds must have sufficient freedom from droop to avoid accumulating excessive error while awaiting readout, which period may be considerably longer than in the case of the converter-per-channel. Increased throughput rate could be obtained by using additional converters, with fewer multiplex switch points and faster update rate.

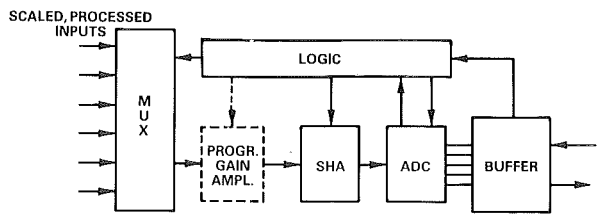
Applications that might require this approach include wind-tunnel measurements, seismographic experimentation, or in testing complex radar or fire-control systems. Often, the event is a one-shot phenomenon, and the information is required in the neighborhood of a critical point during the one-shot event... such as, for example, when a supersonic air blast hits the scale model.

Multiplexing the Inputs of Sample-Holds

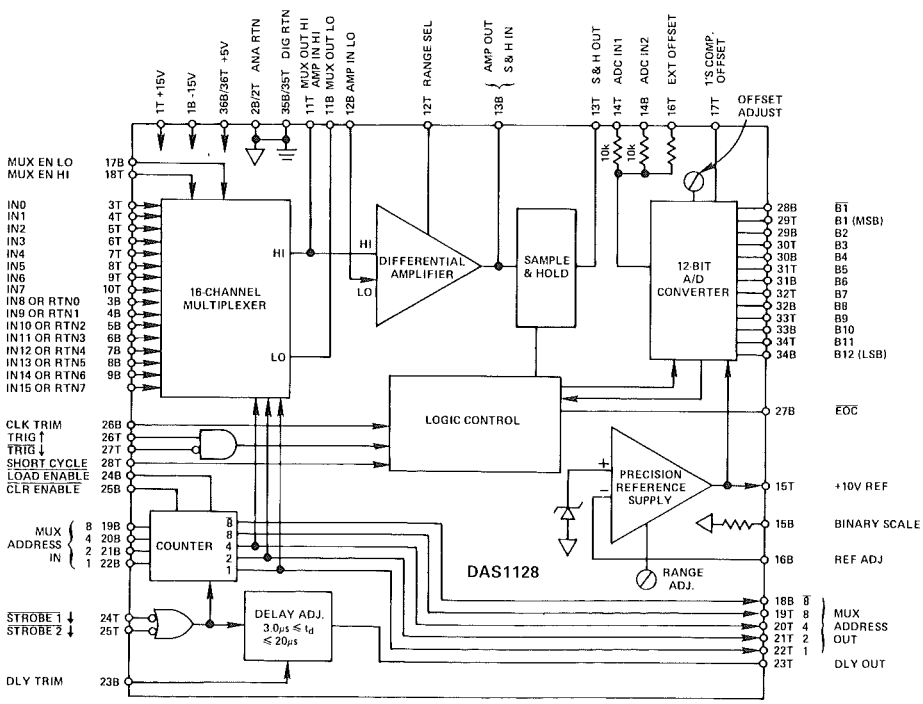
The next step towards increased sharing is to share the sample-hold, as well as the a/d converter. Such subsystems may also include a programmable-gain amplifier for gain ranging (see Figure 2.17). Figure 2.14 shows a basic system embodying this idea, and a block diagram of an early form of device, with wire-programmed range. For most-efficient use of time, the multiplexer is seeking the next channel to be converted, while the sample-hold, in *hold*, is having its output converted. When conversion is complete, the *status* signal

from the converter causes the S/H to return to sample (track) and acquire the next channel. Then, after the acquisition is completed—either immediately, or upon command—the sample-hold is switched to *hold*, a conversion begins, and the multiplex switch moves on.

This system is slower overall than the previous example, and the multiplexer could equally well be switching sequentially or in a random-access mode. For some older systems, a manual mode, for checkout, may also have been used (self-checking and keyboard random access are more typical nowadays). The random-access mode permits channels with more information, i.e., changes per unit time, to be accessed more frequently.



a. Data-acquisition system—basic architecture. The ADC, SHA, and even the PGA (if used) may be combined in one module or IC package to form a single-channel data-acquisition system (sampling ADC). Alternatively, the MUX, SHA, and amplifier are often packaged together to allow a choice of a/d converter.

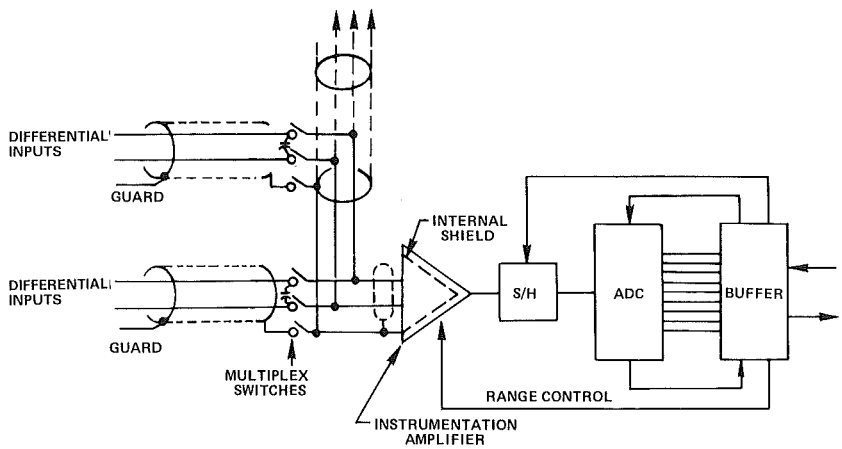


b. Block diagram of early data-acquisition module.

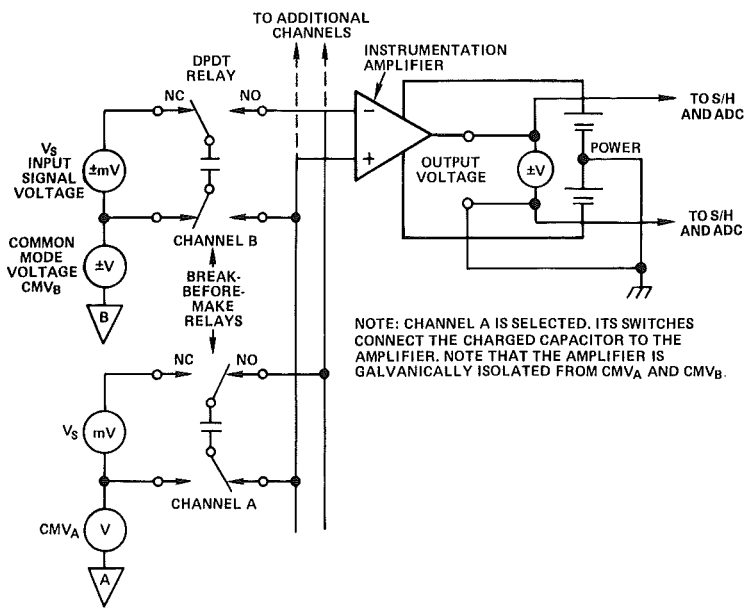
Figure 2.14. Conventional data-acquisition subsystem.

Multiplexing and Signal-Conditioning Low-Level Data

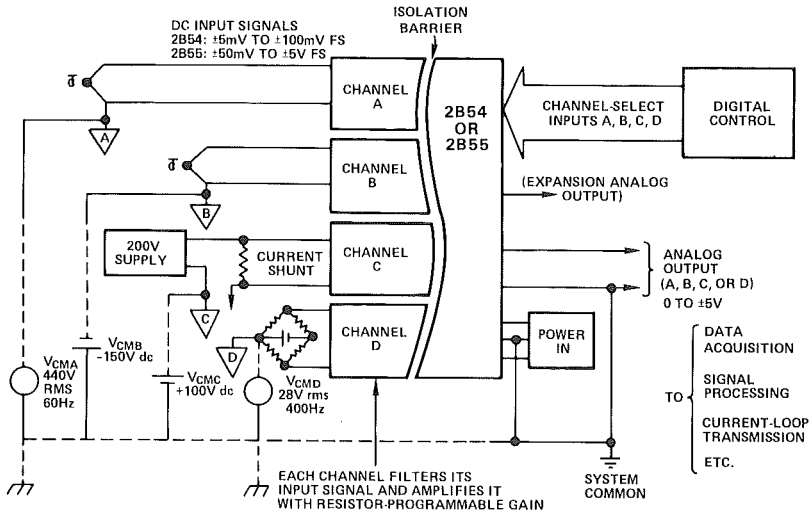
The idea here is that, in addition to sharing of the converter and the sample-hold, expensive signal-conditioning capacity must be conserved. Great strides have been made in recent years in developing effective all-solid-state approaches, supplanting the straightforward "brute-force" approach (Figure 2.15a).



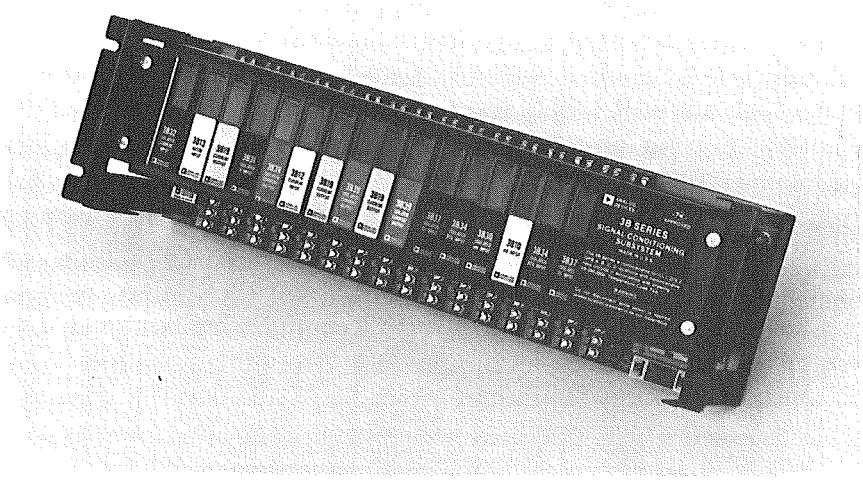
a. Conventional low-level multiplex circuit, showing switched guard. In some systems, two channels are reserved for zero and reference voltage. They are sampled periodically to permit software correction of amplifier offsets and gain errors.



b. Example of flying-capacitor multiplexer.



c. Block diagram of 4-channel multiplexed low-level signal conditioner.



d. A rack-mounting 16-channel signal-conditioning subsystem.

Figure 2.15. Low-level data acquisition.

This has classically been a difficult function to perform. First, the circuitry must be capable of resolving millivolt-level voltages in the presence of large common-mode voltages, with low drift and nonlinearity. The common-mode voltage may be present on the signal's "ground" or it could be induced in the input leads inside a conduit in the vicinity of high-power mains. The signal itself may be afflicted with normal-mode noise.

Safety is a paramount consideration in many of the applications where this kind of circuitry is employed. The input circuitry must be able to withstand

high common-mode voltages without damage, and without exposing the conversion circuitry to high voltage. By the same token, the inputs must be able to withstand the accidental connection of line voltage across any pair of input terminals without mishap to the signal-conditioning circuitry. If the input should be *open-circuited*, due to the failure of a sensor, it would be helpful if an appropriate indication were given, since false information can cause safety problems.

It should be possible to handle signal diversity without introducing complexity, whether the signals are millivolt signals from different thermocouples requiring different scale factors or mixtures of millivolt signals from strain gages and volt-level signals from potentiometers and current-transmitter loops. These conditions are—and have been—difficult to meet at reasonable cost.

Perhaps the most successful approaches are those exemplified by "flying capacitors" and circuits combining solid-state isolation and switching. A flying-capacitor multiplexer is illustrated in Figure 2.15b. Here, capacitors, are switched—generally by relays—between the signal sources and the system input bus; this provides isolation and multiplexing, as well as sample-hold, but it is not easily adaptable to individual gain adjustment. While not necessarily costly in terms of parts, the actual circuit construction requires great care, resulting in a custom installation at high overall cost, with a substantial software overhead for the calibration of individual channels. It is also potentially noisy, and common-mode range is limited by the voltage ratings of the capacitors and switch contacts; speed and life are limited.

An all-solid-state approach, as typically embodied by the Analog Devices Model 2B54 (Figure 2.15c), combines—in a single compact module—transformer-coupled isolation ($\pm$1000V peak max) for each input, differential input protection (130V rms @ 60Hz), signal conditioning, and multiplexing for four channels of millivolt-level input (e.g., from thermocouples). An expansion output provides for the multiplexing of additional groups of four channels *without the addition of external analog switches*. Channels can be scanned at a rate of up to 400 per second, minimum. Accessories are available for connecting sensor inputs directly to screw terminals.

For large numbers of channels having diverse input requirements, input and signal protection, direct connection to field wiring, and standard output formats (both voltage and 4-20-mA current-loop), subsystems are available consisting of powered manifolds of various sizes with assortments of plug-in modules (one per channel) that perform specific signal-conditioning functions. The Analog Devices 3B series (Figure 2.15d) is a typical example of such a system.

2.6 SIGNAL-CONDITIONING TOPICS

Discussed here are a few topics that keep coming up in connection with data-acquisition systems.

Ratiometric Conversion

Some a/d converters have a *ratiometric*, or external reference, connection, allowing the output digital number to represent the ratio of the input to an arbitrary (within specified limits) reference input. In effect, the device becomes an analog divider with digital readout.

Devices of this sort are useful in making precision measurements that ignore variation of a reference used in the measurement. For example, Figure 2.16 shows how a resistance ratio, which might represent a pressure, can be measured—independently of variations of the applied voltage—by applying the same voltage to the reference input of the converter.

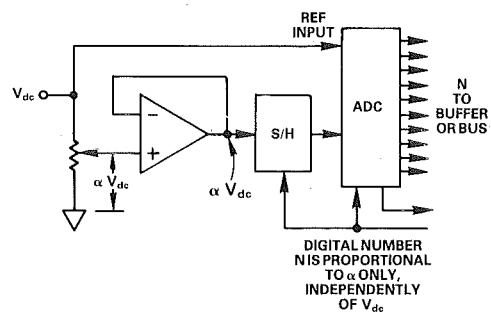


Figure 2.16. Measuring a resistance ratio, independently of the applied voltage—a ratiometric measurement. If common-mode rejection is necessary, the amplifier could be an instrumentation type (or a signal conditioner) instead of a simple single-ended unity-gain follower.

In a multiplexed system, where measurements may be taken from a number of similar devices with a common supply, for example, strain-gage bridges, the common bridge supply may be used as the converter's reference to eliminate normal-mode gain error caused by supply-voltage (or converter reference) variation.

Wide Dynamic Ranges (see also Chapter 17)

The need for wide-dynamic-range signal conditioning in a single channel may occur in two basic ways: Either it is necessary to resolve a voltage anywhere in the range to a high degree of accuracy, relative to full scale (for example in the measurement of position in a followup system); or it is sufficient to measure a quantity having a wide range of variation to modest accuracy, relative to actual value (for example, to within 1%, over a 10,000:1 range).

For signals in the first category, a high-resolution-and-linearity converter is the simplest answer.

To maintain wide dynamic range for small signals, it is feasible to use a moderate-resolution converter (say 12 bits), preceded by a software-programmable-

gain amplifier, i.e., an amplifier with high-accuracy switched gain controlled from the digital interface. Software-programmable-gain amplifiers are available in several forms: They can be found in data-acquisition subsystems (Figure 2.17); purchased as complete modules or hybrids; and assembled from

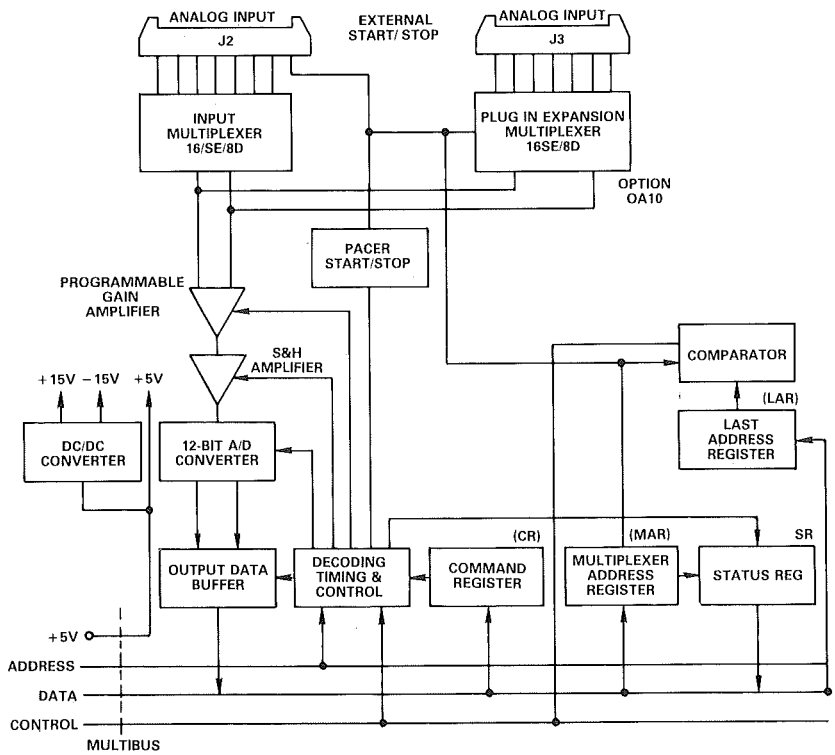


Figure 2.17. 1-2-4-8 programmable-gain amplifier extends converter dynamic range. Data-acquisition portion of the RTI-732 MULTIBUS-compatible analog I/O subsystem.

high-performance instrumentation amplifiers with external resistors and CMOS switches (Figure 2.18).

In one form of operation, a trial conversion is performed at the lowest gain; if the MSB is 0, the gain is doubled and another conversion is performed; if the MSB is still 0, the gain is doubled again, etc., until either the MSB is turned on or the top of the gain range is reached. Each doubling represents an additional bit of resolution for small signals. The scheme shown in Figure 2.17 is employed in a MULTIBUS-compatible analog I/O subsystem (see Chapter 4), for up to 15 bits of dynamic range and 12-bit resolution and accuracy.

Yet another possibility, when seeking accurate measurements of small variations about a fixed value of voltage, is to take the difference between the input and an accurately set voltage equal to the nominal fixed value. If the voltage

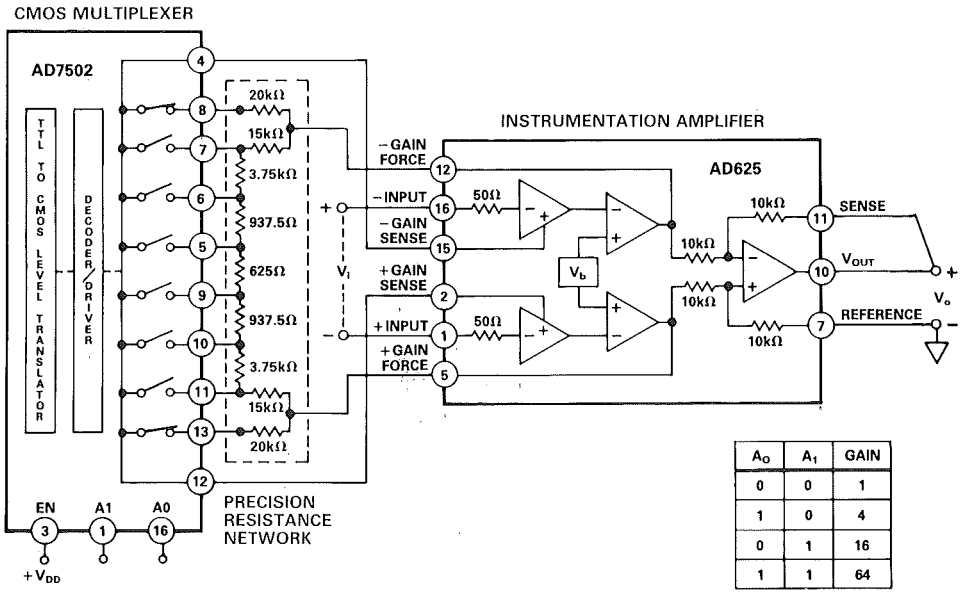


Figure 2.18. Software-programmable-gain amplifier.

is applied via a high-resolution DAC (Figure 2.19), the interface can keep track, digitally, of both the initial value and the difference voltage, using an ADC of relatively modest performance. (The tradeoff here is the cost of a high-resolution DAC, plus logic and a modest 8-, 10-, or 12-bit DAC, vs. a 16-bit ADC.) This configuration also forms the basis of an excellent ADC test scheme (see Chapter 10).

Another approach to handling wide dynamic ranges with converters having limited resolution is to compress the data through the use of logarithmic techniques, in the form of either logarithmic converters or logarithmic processing of the analog signal (Figure 2.20). Logarithmic converters will be discussed in Chapter 16.

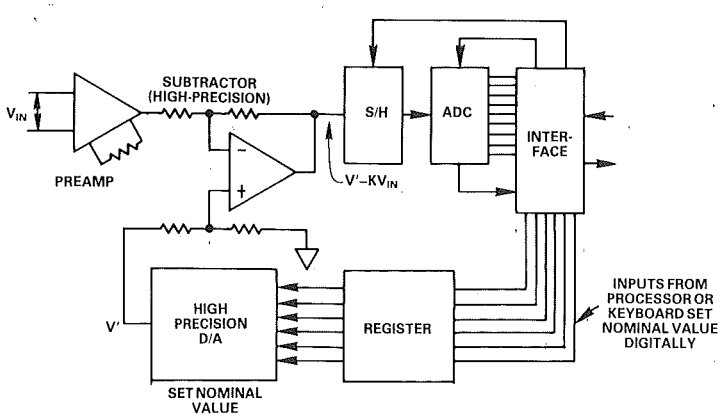


Figure 2.19. Use of a high-resolution DAC to measure small deviations about a precisely determined value.

The error of a logarithmic amplifier, after calibration, is a *log conformity* error (nonlinearity on a semilog plot), specified in terms of a maximum value at the output, or a maximum ratio to actual input over a specified range. For example, 1% log conformity error means that the error at the output, for 2V/decade scaling,* is 8.6mV, corresponding to an input uncertainty of ±1%. Typical input voltage range (e.g., for Analog Devices Model 755) is 1mV to 10V. The corresponding output voltage range is ±4V (i.e., ±2 decades at 2 volts per decade, with respect to a 0.1-V reference level).

Since an error of 1%, referred to the 1-mV minimum input signal, is $1/10^6$ of full-scale input, and since the corresponding output error of 8.6mV is $0.0086/8 = 1.075 \times 10^{-3}$ of the output swing, the dynamic range of the signal has been compressed by a factor of 1,000, as a result of the logarithmic transformation. This means that a 12-bit converter (with suitable scaling) can be used to digitize the log-amplifier output, with a quite comfortable error margin.

Though it might appear that the representation of data having an inherent 20 bits of resolution ($10^6 \cong 2^{20}$) by a signal having 12 bits of resolution is getting "something for nothing," in violation of *some* Natural Law, the scheme really works. There are, however, some points to consider:

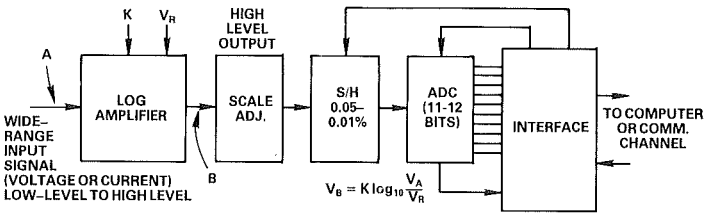
1. Compression is achieved by exponentially distorting the relative value of the least-significant bit. Thus, for a 10,000:1 signal range, represented by ±4-V output, an LSB (of 12 bits, offset binary, suitably scaled) is worth about 23mV at 10V input (i.e., $0.1[\exp_{10}(2 - 8/8192)] - 10V$ and 2.3μV for 1mV input. Therefore, while the approach is quite useful for compressing data requiring essentially constant *fractional* error (e.g., 1%) anywhere in a wide range, it is not at all suited to applications requiring high resolution (e.g., 0.01%FSR) *at any point* in the range.

2. Since the digital number is a logarithmic representation of the analog input signal, it must be dealt with as such in the digital process. If the number is to be used in computation, it should be antilogged, using either a lookup table or processor computing capacity—unless, of course, the computation is facilitated by the availability of a logarithmic relationship. If the data is to be stored or transmitted, and eventually returned to analog form unaltered, it does not require any further digital transformation, just an analog antilog operation following the output d/a conversion (unless logarithmic analog data is desirable).

3. Since a logarithmic function is inherently unipolar (the logarithm is real only for positive values of the argument—positive signals require a 755N, negative signals a 755P), it is far from ideal for signals that are inherently zero-centered. While it may be useful to bias some types of input signals into a single polarity, functions that demand symmetrical treatment may be badly

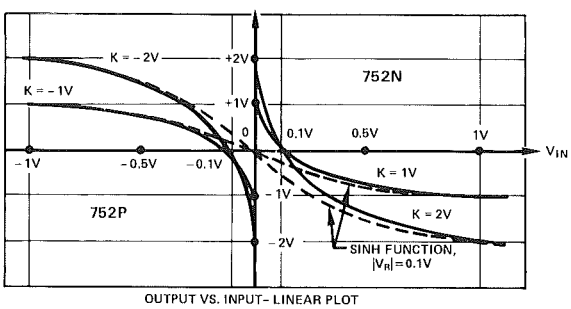
*A *decade* is a 10:1 range of input voltage or current.

distorted by the wide variation, in both resolution and speed, between zero and full-scale input. Such functions would profit by a type of compression that is symmetrical about zero. An example of an easily obtained form is a $\sinh^{-1}$ function (Figure 2.20), which involves two complementary antilog transconductors (752P and 752N) in the feedback path of an op amp. The resulting function is logarithmic for larger values of input, but it passes through zero, essentially linearly (and slowly).

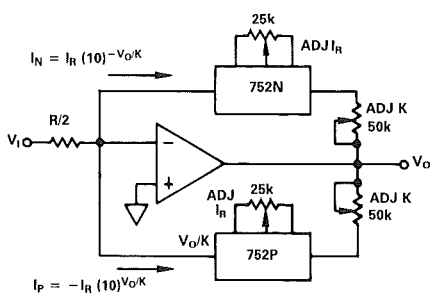


A: INPUT, e.g. 10V TO 1mV ±1% (10μV RESOLUTION AT LOW END)
B: OUTPUT, ±4V ±8.6mV (K = 2, V_R = 0.1V) (RESOLUTION ABOUT 0.1%)
IF K ADJUSTED TO MATCH ADC, "SCALE ADJ" NOT NECESSARY

a. Log amplifier for range compression in a data-acquisition system.



OUTPUT VS. INPUT- LINEAR PLOT



$$2\frac{V_I}{R} = I_N + I_P = I_R (10)^{-V_O/K} - I_R(10)^{+V_O/K}$$

$$-\frac{V_I}{I_R R} = \frac{\epsilon^{2.3V_O/K} - \epsilon^{-2.3V_O/K}}{2} = \sinh\left(2.3\frac{V_O}{K}\right)$$

$$-V_O = \frac{K}{2.303}\sinh^{-1}\left[\frac{V_I}{I_R R}\right]$$

b. Bipolar signal compression using complementary log transconductors to synthesize the $\sinh^{-1}$ function.

Figure 2.20. Logarithmic amplifiers in data acquisition.

Noise Reduction

Like diseases, noise is never eliminated, just prevented, cured, or endured, depending on its seriousness and the costs/difficulty of treating it.

Analog noise in data-acquisition systems takes three basic forms: *transmitted noise*, inherent in the received signal, *device noise*, generated within the devices used in data acquisition (preamps, converters, etc.), and *induced noise*, "picked up" from the outside world, power supplies, logic, or other analog channels, by magnetic, electrostatic, or galvanic coupling.

Noise is either *random* or *coherent* (i.e., related to some noise-inducing phenomenon within or outside of the system). Random noise is usually generated within components, such as resistors, semiconductor junctions, or transformer cores, while coherent noise is either locally generated by processes, such as modulation/demodulation (e.g., chopper-stabilization), or coupled-in. Coherent noise often takes the form of "spikes", although it may be of any shape, including—collectively from many sources—pseudorandom.[4]

In systems involving the conversion of analog signals, the finite resolution of the conversion process introduces "quantization noise," which may be thought of as either a truncation (or roundoff) error, or as a noise, depending on the context. See also Chapter 17.

Noise is characterized in terms of either *root-mean-square (rms)* or *peak-to-peak* measurements, within a stated bandwidth.* Random noise from a given source, within a given bandwidth, will give consistent *rms* measurements. For a typical gaussian amplitude distribution, and a sufficient number of measurements, one may expect a consistent relationship between the probabilities of obtaining peaks of a given size in relation to the rms, as shown in the Tables in Figure 2.21.

RMS values of noise from uncorrelated sources (e.g., from different devices, or from different portions of the frequency spectrum of the same device) add as the square-root of the sum-of-the-squares, and if the largest is more than 3 times as large as any other, the others may usually be safely ignored. However, if noise is dominated by picked-up spikes, root-sum-of-squares is of small comfort.

As we have indicated at the beginning of the chapter, there are ordinarily two basic forms of system-design problem: those involving essentially ordinary signal levels in unfavorable environments, and those involving extremely high-resolution measurements in favorable environments. (However, our readers should be aware that Murphy's Law would imply that their system design problems tend mostly to involve high resolution measurements in unfavorable environments.)

*For further references to noise, see the Bibliography.

[4]"Understanding Interference-Type Noise," *Analog Dialogue* 16-3 (1982), pp. 16-19.

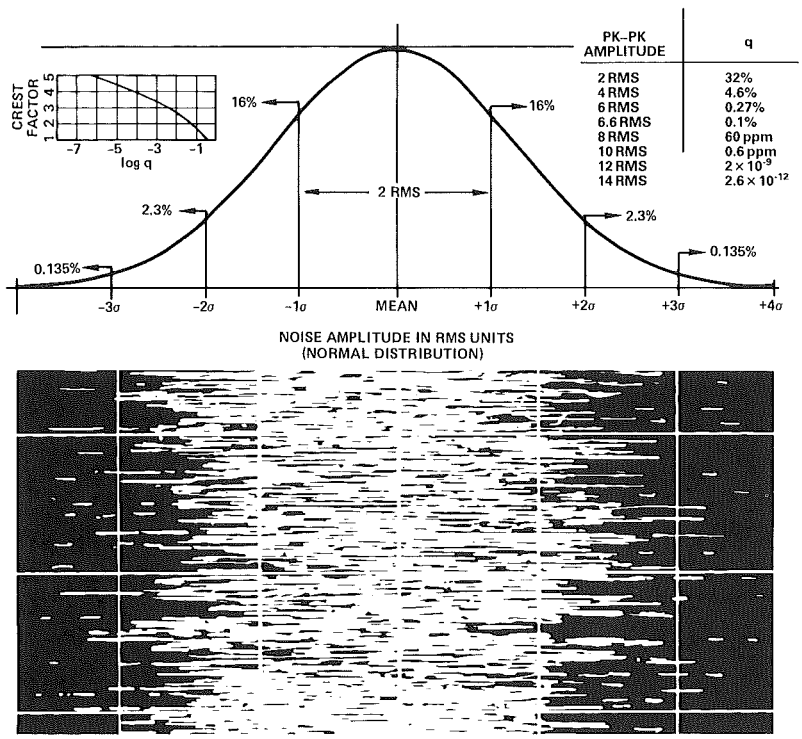


Figure 2.21. RMS vs. peak-to-peak amplitudes for gaussian noise.

For unfavorable environments, where the major sources of noise are *induced noise*, the designer must rely on early preamplification and conversion, isolation, shielding and guarding, signal compression and filtering, and—where possible—an information rate (via digital filtering, fast sampling or parallel paths) that has enough redundancy to allow the digital processor to retrieve data via digital filtering, correlation, and summation (see Chapters 6 and 21).

In favorable environments, where the measurement process and the processing hardware introduce the major portion of the uncertainty, the emphasis must be placed on measurement techniques, filtering, choice of data-acquisition hardware for best resolution, and—again—the use of high-speed digital processing for signal retrieval plus "intelligent" measurement techniques including automatic drift compensation and scale-factor adjustment.

Where noise is likely to have large spikes as a major component, the integrating-type converter usually provides additional filtering. For random noise, if there are sufficient samples taken of a given signal channel, the statistical properties of the noise are imparted to the digital output, which may be filtered by digital techniques.

Chapter Three

Data Distribution

After analog data have been converted to digital form and have been duly stored, transmitted, or processed, the results of this handling, as well as some newly created digital numbers, may be required once again to intervene in the "real world" of phenomena. In analog or digital form, they may be used to drive meters or motors, display information, stimulate devices under test, generate heat, light, or sound, modulate waveforms, sound the alarm, adjust an audio gain, or—in short—to manipulate energy according to a digital code.

The digital output words are made available fleetingly on an output bus—or for longer periods, at an output register—for distribution to their destinations. While an increasing number of real-world devices, such as numerical displays, stepping motors, printers, and the like, are operated more or less directly by digital numbers (perhaps with "decoding," but without the overt interposition of electronic analog variables), there is a widespread—and growing—use of electronic d/a converters in the redeployment of digital data in analog form. This chapter treats of systems that use d/a converters.

As with a/d conversion, but reversing the order, the basic objective is to get the data into the appropriate analog form, as rapidly, as frequently, as accurately, as completely, and as cheaply as necessary.

The basic instrumentality for accomplishing this is the digital-to-analog (d/a) converter (DAC). In response to a digital code, it may be used either to provide a voltage or current output (fixed-reference DAC), or to adjust the gain of an analog circuit (multiplying DAC). It can be a simple device on an IC chip, or a sophisticated high-resolution high-speed device with many "bells and whistles;" physically, it may take the form of a box, a card, a potted module, an integrated circuit—or even a portion of an integrated circuit. It may

be functionally integrated with other system elements to form a subsystem.

To accommodate the analog output to the specified conversion relationship, some form of scaling and offsetting (signal conditioning) and energy translation (e.g., current-to-voltage) may be necessary, performed with amplifiers. To furnish analog information to more than one destination, either additional converters or a multiplexer and sample-holds may be necessary. To encompass an extra-wide analog dynamic range, an exponential amplifier or conversion relationship may be found useful.

The nature of the data-distribution system depends on the properties of both the digital and analog data, and what is to be done with it. This chapter deals with aspects of signal flow, from the digital data source through conversion. Chapters 4 and 6 have to do with integration into systems and instruments, Chapter 7 deals with the fundamentals of DACs, Chapter 8 deals with some of the forms IC DACs can take, Chapter 13 deals with DACs designed specifically for high-speed ("video") applications, Chapter 16 considers (intentionally) nonlinear DACs, and Chapter 17 discusses high-resolution converters.

Commercially available data-distribution systems range from basic d/a converters to multi-channel converters on monolithic chips to completely integrated systems on cards and in boxes, and even include converters that are functionally inseparable from the digital processor. In later chapters, we will consider the various optional levels of integration available in a single package or piece of equipment. However, in this chapter, we will generally treat most of the functions peripheral to the d/a converter *as though they were embodied by separate components*, in order to make clear the architectural choices that a designer might have, with their characteristics, advantages, and disadvantages.

3.1 FACTORS AFFECTING DISTRIBUTION-SYSTEM DESIGN

The configuration, choice of components and their specifications, the system timing, and location of multiplexing, depend, as with data acquisition, on

1. Number of channels
2. Update window per channel
3. Update rate
4. Bus width and word length
5. Output resolution
6. Output linearity and accuracy
7. Settling time per channel
8. The nature of the loads
9. The cost function

There are a number of additional areas for decision by the system designer:

Digital signal source: Parallel bus? port? register? serial data—bit-serial, byte-serial, ASCII?

Signal storage between or during updates: External or internal registers? Single-rank, dual-rank, n-rank? Analog storage (sample-holds, inertia)?

Multiplexing: μP bus? Digital switching? Multi-channel DACs? Analog multiplexing (sample-holds or multiplex switches)?

Update: Simultaneous? Sequential? Random?

Conversion: Near digital source or remotely? Many DACs—or few DACs with multiplexing?

Analog Output: Voltage, current, or gain? Discrete values or smoothed? Permissible level of switching transients, use of deglitching? Direct-wired or galvanically isolated circuitry?

Cost tradeoffs: Minimizing use of expensive components. Sample-holds vs. multiplexers vs. DACs. Inertial filtering. Using low-precision incremental "slave" DACs for accurate settings.

3.2 DIGITAL SIGNAL SOURCE

The basic d/a converter accepts parallel digital data at its input and continuously provides a representative analog output, usually based on a binary relationship. As soon as the digital code changes, the analog output seeks to follow it and—after a transient interval, of variable turbulence—comes to rest at the new value, within a period ranging from nanoseconds to microseconds.

Since there is a continuous input-output relationship, if a basic DAC must maintain a constant output after updating, it must be fed a continuous digital input. However, unless the DAC is wired so that its inputs are totally dedicated to a unique flow of information (for example, from a counter, as in Figure 3.1, or from a dedicated computer output port), it must get its inputs from a source which is rapidly changing and at the same time servicing other I/O (input/output)—or even internal—elements in the system.

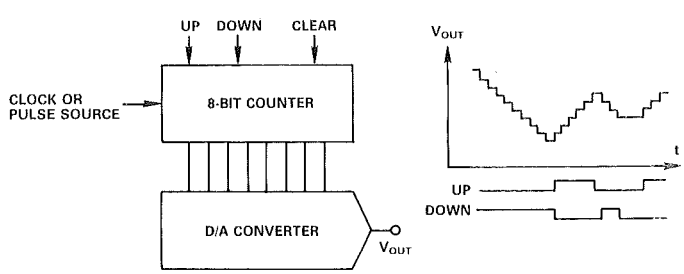


Figure 3.1. Converter output is a continuous analog representation of the counter's digital output (simplified block diagram).

3.3 REGISTERS

The d/a converter's input, therefore, almost always comes from a register, which is latched upon command ("clock" or "strobe") to preserve a fixed

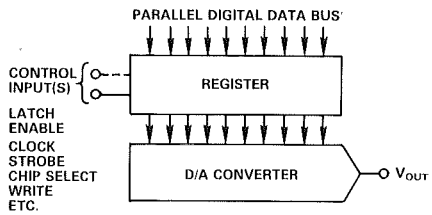
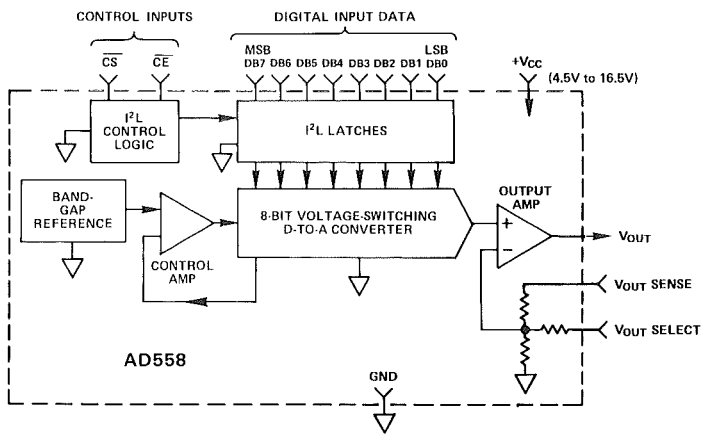
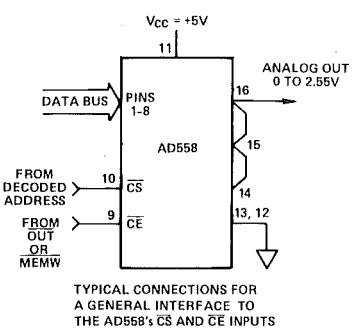


Figure 3.2. Basic d/a converter is buffered by a register. When it is latched, the output voltage is unaffected by the data on the bus.

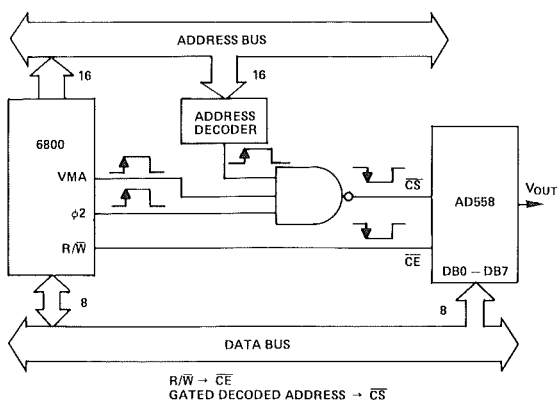
digital code (Figure 3.2). For a single-stage latch, the output will follow the input while the control signal is in one state (transparent) and become latched when it changes state; for a two-stage latch, the output ignores the new value of input that is being acquired until it is latched in when the control signal returns to the inert state. The register may be external to the DAC chip (or package) but is often contained within it.



a. Voltage-output monolithic 8-bit DAC.



b. Control signal connections.



c. Interface to a 6800 microprocessor.

Figure 3.3. Interfacing an 8-bit DAC to a microcomputer bus.

The register is controlled by one or more latching signals. In microprocessor systems, the conjunction of a decoded addressing "chip select" and an enabling "write" command—within the window of time that valid data is on the data bus—will cause that data to be latched. Figure 3.3 shows a typical latching scheme used for a completely self-contained voltage-output monolithic 8-bit DAC having a self-contained register. The block diagram of the DAC is shown in (a), typical connections in (b), and the interface to a 6800 microprocessor in (c).[1]

The *chip select* input goes low when the clock ($\phi 2$) goes high, VMA (Valid Memory Address) goes high, and the address decoder selects the device in question by going high; the *chip enable* input goes low when R/$\overline{W}$ goes low to indicate that data is to be written to the device. When both are low, the DAC's register is "transparent" and open to updating by the data on the bus; as soon as either returns high, the data is latched. After one or two microseconds, the analog output has settled to its new value.

3.4 MULTIPLE RANKS

More than one rank of registers may be employed, in a manner determined by the way the data is arriving from the signal source. For example, if the data is placed on the bus at the convenience of the processor, but it is not yet time to update the DAC, the data may be latched by the first rank. It will be loaded into the DAC register (second rank), when the time arrives to update the DAC, either synchronously, as determined by the processor, or by some other (asynchronous) entity.

If all the data does not arrive simultaneously in parallel, two or more stages of latching may be required to prevent false data from appearing at the analog output. Figures 3.4 through 3.6 show examples of situations calling for more than one stage of latching.

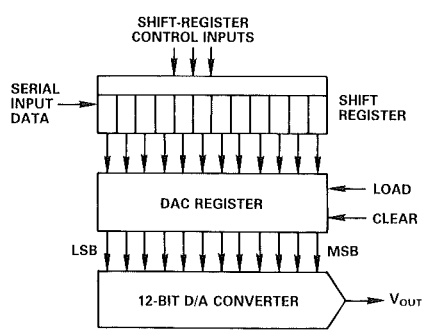


Figure 3.4. 12-bit DAC with serial input. Data is clocked into the shift register in serial (old data latched in DAC register), loaded into the DAC register in parallel at will.

[1]See "Putting the AD558 DACPORT™ on the Bus," by D. Grant. *Analog Dialogue* 14-2, 1980, pages 16-17.

In Figure 3.4, 12-bit data arrives—MSB first—at the serial input (SRI) of a DAC designed to accept bit-serial data. At a time when each bit can be expected to be stable, it is strobed into the shift register. The process continues, bit by bit, until all bits have been loaded, without affecting the DAC output. After all bits are loaded, the DAC register can be loaded at any time, to update the DAC output. The scheme is analogous to a 12-car train pulling into a station, car by car, then stopping and opening all the doors at once to detrain the passengers.

12-bit data generated for distribution on an 8-bit bus must be delivered as two separate bytes. The bytes can be presented in two ways, either "right-justified" or "left-justified." In the former scheme, with the data written in the fashion of whole numbers, the 12-bit word, 0100 1000 0001, would consist of two bytes, 0000 0100 and 1000 0001; in the latter scheme, written as a binary fraction, it would be 0100 1000 and 0001 0000. In either case, the two bytes must be simultaneously available at the input of the DAC register in order to obtain a correct analog output. Figure 3.5 shows the block diagram of a 12-bit DAC (the AD667), which accepts data in three four-bit nybbles, and the addressing scheme that would provide a left-justified 8-bit bus interface.

The 8 most-significant bits of the first rank, DB11 through DB4, are wired to the data bus, and the last four bits (the top four of the second byte) are wired to DB11 through DB8. When $\overline{WR}$ goes low, the first 14 bits of the address, A15 through A2, are decoded to enable the device, via the Chip Select input; when the last two digits of the address are 01, the 0 at A1 enables the last four bits, which are subsequently latched; then, when the last two address digits are incremented to 10, the 0 at A0 enables the 8 most-significant bits and at the same time loads the entire word into the DAC latch, updating the DAC's output.

It is easy to see that, with the facilities it has available, the same DAC could also be used to update the entire 12-bit word at once from a 12- or 16-bit bus, to update the word in three nybbles from a 4-bit bus, or to provide a right-justified 8-bit bus interface.

The forms of interfacing described above are often called *memory-managed interfacing*, since the d/a converter is communicated with in the same manner as a memory location (see also Chapter 4); it looks like a *write-only* memory. There also exist DACs with *readback* capability; they make the word that is currently providing the DAC output available to the bus in response to a *read* command—a helpful capability that makes it unnecessary for the processor to continually remember the last value, an especially useful feature at startup or after interruptions.

3.5 MORE THAN ONE CHANNEL

Since many I/O entities may be connected in parallel on the data bus, with each responding to a unique address code, digital multiplexing is inherent

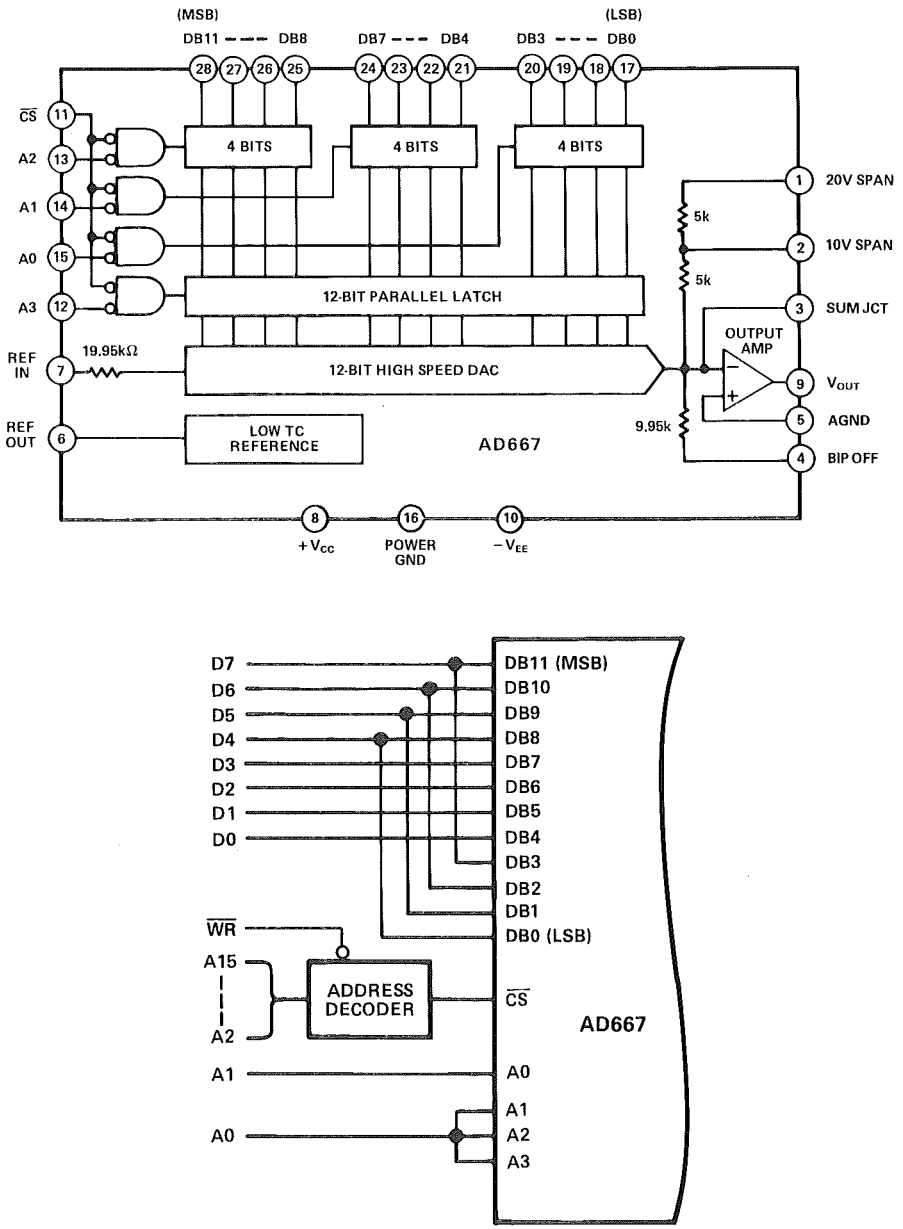
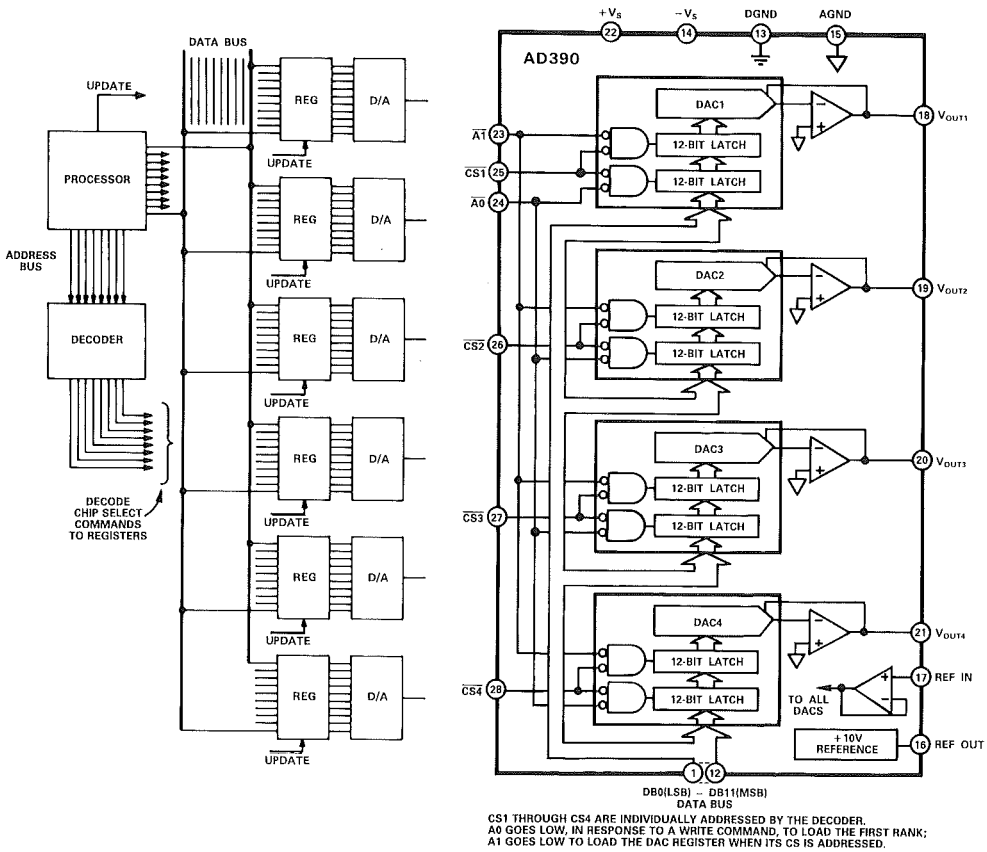


Figure 3.5. Double-buffered 12-bit DAC capable of interfacing with 4-, 8-, 12-, and 16-bit data buses. Left-justified connection to 8-bit bus.

(Figure 3.6a). If a number of DACs are connected to the bus, only those that are addressed will receive an update via their associated register(s) when the WRITE command is received. The converters may be addressed randomly (and repeatedly) or in a sequence. It is good practice to use two-stage latches or double buffering so that the basic DAC is never exposed to bus noise.

The multiple DACs may all be in the same package. For example, the quad DAC shown in Figure 3.6b contains four 12-bit double-buffered voltage-output DACs multiplexed on a 12- or 16-bit data bus.



a. Converters share a common bus.

b. Quad DAC: 4 12-bit DACs in a single hybrid package sharing common bus.

Figure 3.6. Converter-per-channel distribution.

Digital multiplexing may also be performed with multipoint switching: For example, if four n-bit DACs with parallel inputs are to be updated, each bit is switched to one of four output lines (Figure 3.7). Though more complex, this approach has the advantage of reducing the load on the bus, since only the DAC currently being addressed is connected.

If there are many analog channels, and sample-holds are less costly than DACs for a given speed and resolution, the designer has the option of using a single d/a converter—multiplexing its output among many sample-holds, either with a multiplexing switch or by feeding the analog input to all of the sample-

holds in parallel and decoding the address line to the sample-hold control inputs (Figure 3.8). A two-channel application of this technique might be found in digital stereo decoding, where a single 16-bit DAC updates the two audio channels alternately via sample-holds (see Chapter 17).

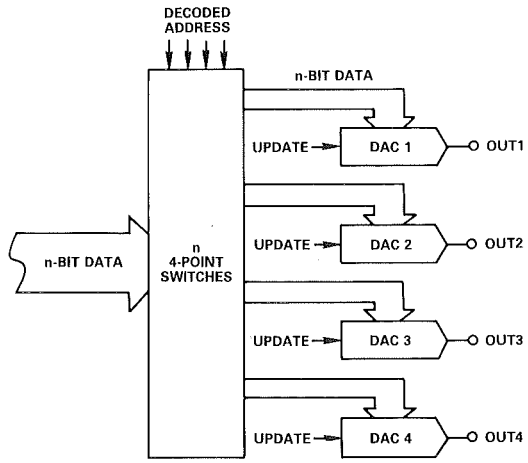


Figure 3.7. Multipoint switching of four d/a converters.

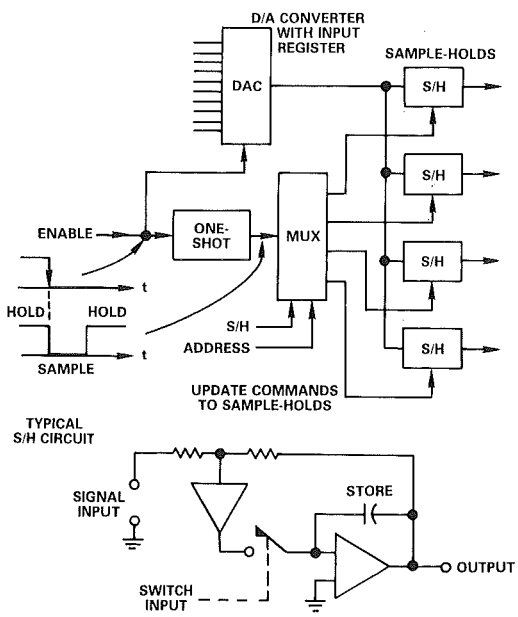


Figure 3.8. Sample-holds with common analog inputs and multiplexed control inputs. Desired sample/hold is addressed. Then, on next update, DAC is latched, and addressed S/H is switched to SAMPLE. One-shot switches it back to HOLD when sufficient time for settling has elapsed.

Comparing Sample-Holds and DACs

Once updated, a d/a converter-with-input register will store an analog value indefinitely, or at least for so long as the power is connected. By contrast, a sample-hold circuit, since it holds the analog data on a capacitor, is susceptible to a definite "droop" (positive or negative) in the analog output as the charge on the capacitor changes due to leakage across the switch, from the amplifier's summing point, or from the supplies (or perhaps even due to the capacitor's own leakage resistance or dielectric absorption).

Thus, even though the data may not change at all, if the same value must be maintained for a long period of time, it is necessary to update the sample-hold periodically to correct for output droop. On the other hand, so long as the data remains unchanged, a distribution system based on d/a converters (with registers) has no need to be periodically refreshed. In fact, the DAC's ability to store without error lays the foundation for saving time by "updating by exception," whereby the data channels are updated only if the data changes.

A further consideration in the use of d/a converters vs. sample-holds lies in the matter of allowing for acquisition and settling time. The data sheet for a typical general-purpose data-distribution sample-hold circuit at reasonable cost may call for acquisition periods ranging from 1 μs to 26 μs or more. Thus, the multiplexer must dwell at each channel for the duration of this acquisition period, and the update sequence must be arranged to avoid tying up the processor bus for long periods.

Offsetting some of the speed and flexibility of the DAC-per-channel method is the cost of interconnecting the DACs to the data source. Parallel data at the 10-bit (0.1%) level requires at least 12 conductors (10 data lines, common return, and command line). If the d/a converters are at any distance from the bus, or its buffer, installation cost for the cable may become the largest single economic factor, far outweighing the cost of the DACs. Cable and installation costs can be greatly reduced by introducing serial (bit-at-a-time), instead of parallel, transmission, but at a considerably reduced update rate.

In addition to allowing asynchronous timing of the loading and analog update for each DAC, systems employing double-buffered d/a converters permit simultaneous updating of a number of DACs. The first rank of registers is loaded as each item of data is made available, then the second ranks of all devices (the DAC registers) are updated simultaneously. (This is similar to the scheme used for a DAC that must accept data from the bus in more than one byte but must update the DAC register simultaneously.)

Analog Data Distribution

One of the approaches to sample-hold-circuit updating is shown in Figure 3.8. Analog data is sent over a common wire to all the sample-hold circuits. However, each s/h, normally in *hold*, remains oblivious to the input data until a command signal connects it momentarily to the analog data bus (*sample*). On

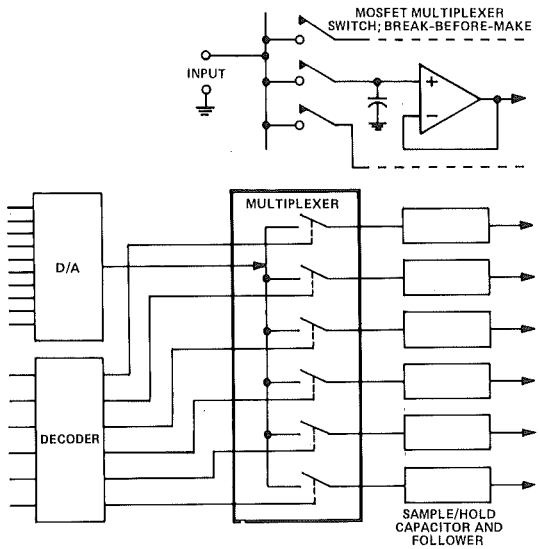


Figure 3.9. Sample-holds with multiplexed *analog* inputs. Multiplexer switch can do double duty in simplest configuration.

receipt of its update pulse, the sample-hold circuit acquires whatever analog information appears on the data line and holds this value until subsequently commanded to acquire a new signal level.

An alternative arrangement, potentially low in cost (and fraught with challenges for the designer), uses an analog multiplexer for distribution of analog data among individual channels, as shown in Figure 3.9. Here, the sample-holds respond to whatever signals are presented at their input terminals, and then hold this signal level when the analog input is disconnected. The multiplexer's switches serve double duty, both in multiplexing and as an interruptible path for charging the hold capacitor, though more complex switching arrangements may be used. It is more subject to leakage and crosstalk than the circuit of Figure 3.8, and requires some care in the timing of switch operations; but the idea is conceptually simple and low in cost.

3.6 ACQUISITION vs. DISTRIBUTION

As a rule, data *acquisition* poses more challenging problems than data *distribution*, but some of the problems assume different shapes. Since data distribution can take place at macroscopic power levels (volts and milliamperes), noise is not a great problem (except for induced noise in hostile environments). To the contrary, DAC outputs may be boosted, as in programmable power supplies; in such cases, it is useful for the DAC's output amplifier system to have remote sensing (force-sense, or Kelvin connections) to avoid errors due to voltage drops in the wiring. This is also good practice for high-resolution (16-18-bit) DACs, even at more-modest power levels.

Sample-Holds used in data acquisition must have short aperture time (or at least small aperture uncertainty) because they must either deal with the "instantaneous" value of a signal, or sample it rapidly at equal time intervals. Their *hold* time need be no longer than is necessary for the ADC to digitize the signal. In short, the usual emphasis in sample-hold circuits for data acquisition lies on rapid acquisition, followed by rapid conversion.

By contrast, sample-hold configurations used for data distribution usually permit relaxed update timing, but the analog values may have to be preserved for long periods without significant *droop*. Thus, sample-holds for distribution must have long *hold* times, and short acquisition-and-settling times. Where high resolution (12 bits or better) and large ratios of *hold* to *settling time* are necessary, multiple-DAC distribution—with register storage—becomes preferable; the decreasing cost of IC DACs makes the choice an easy one.

3.7 FILTERING AND DEGLITCHING

In data acquisition, analog filtering is used to remove (or at least reduce) analog transmitted, inherent, or induced input noise and to eliminate components of signal and noise at frequencies greater than one-half the sampling frequency to avoid aliasing. In distribution, filtering is used to reduce "noise" caused by quantization and sampling (finite increments of digital resolution and discrete output values due to sampling cause discontinuous analog outputs, which introduce unwanted frequency components) and to deal with coupled-in switching transients.

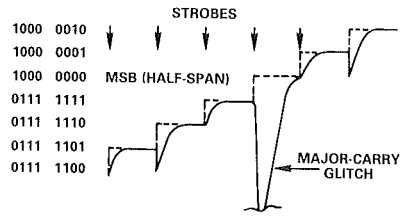
Small discontinuities are often tolerable, especially in dc-value testing; they occur at the application of test conditions, and readings are not taken until the system has settled. On the other hand, if the converter is producing an analog ramp in discrete steps, the discontinuities may have to be smoothed, and certainly any feedthrough transients and/or "glitches" must be minimized.

For reconstructing coarse sampled data, sophisticated analog interpolation techniques are used to overcome the limitations of simple filtering. An example is integration of the difference between two adjacent values so that the "points" are connected by straight lines or exponentials, and discontinuities become more-easily filtered changes in slope rather than steps.

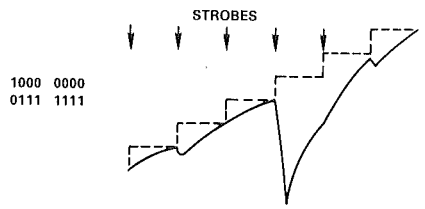
A "glitch" is an insidious spike caused by intermediate codes introduced by asymmetrical switching times at major-carry transitions, such as from 0111 1111 to 1000 0000. In this example, where the DAC output is to change by one least-significant bit—from 1LSB below half scale to half scale—if the less-significant bits all switch off slightly before the MSB switches on, the DAC output will momentarily seek to go all the way towards zero (Figure 3.10a), then return to half-scale, creating a very large spike.

Linear filtering of glitches is impractical, because they have far-from-uniform magnitudes, and they do not occur at uniform intervals; hence, linear filtering

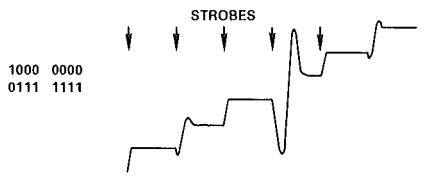
can lead to badly distorted waveforms (Figure 3.10b). Glitches are minimized in high-speed "minimum-glitch" DACs by the use of latches and very fast switching, with the best-possible matching of rise and fall times, so that (if possible), the remaining error is a small, fast, filterable doublet pulse with near-zero average energy (c).



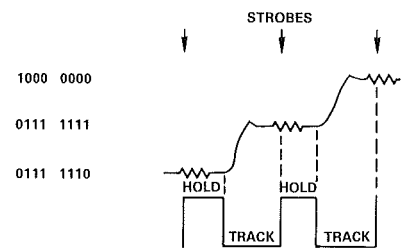
a. Transitions in the vicinity of the major carry, showing glitches.



b. Filtering's distorting effect at the major carry.



c. Typical response of DAC designed for minimal glitch.



d. Effect of track-hold deglitching. Note slower update rate on same time scale as above.

Figure 3.10. Glitches in d/a converters.

In the more-usual case, track-hold *deglitcher* circuitry is used to cause the DAC's output circuit to ignore the glitch. The output circuit is switched into HOLD while the DAC is updated, then switched back to TRACK after a sufficient time (perhaps established by a one-shot) has elapsed for the glitch to settle out (Figure 3.10d). The deglitcher circuitry, though cleaning up the response, will result in a reduction of the update rate. Figure 3.11 is the block diagram of a complete 12-bit deglitched DAC capable of a 6-MHz update rate.

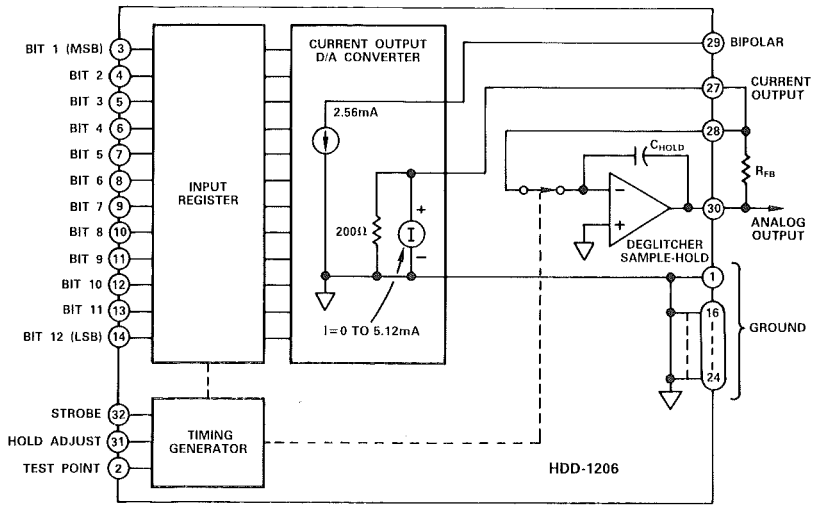


Figure 3.11. Block diagram of 12-bit deglitched DAC capable of 6-MHz update rate. Gain is set by external feedback resistor.

3.8 MINIMIZING CALIBRATION ERRORS BY SERVOING

In a test system, it may be necessary to set a number of parameters with high resolution and accuracy. It can be done using high-performance DACs, but where many channels of accurate analog output are required, it is possible to accomplish this at lower cost with a single high-performance DAC and sets of paired lower-performance converters. A representative scheme is shown in Figure 3.12.

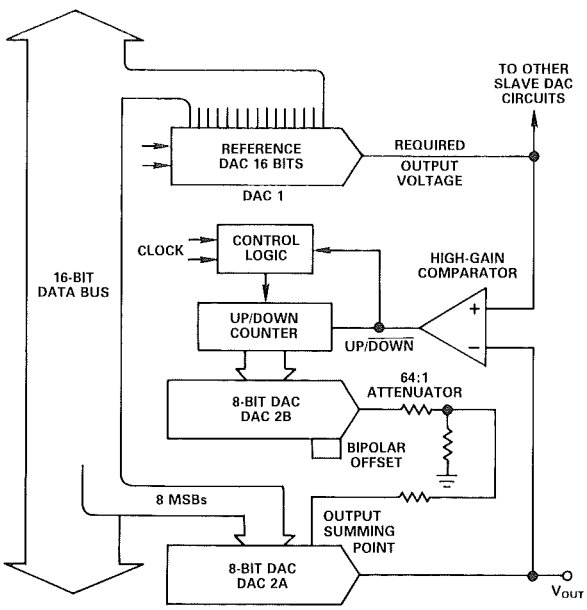


Figure 3.12. Master-slave precision d/a converter scheme, using 16-bit master and two 8-bit DACs per channel for 14-bit resolution, employing comparator and counter for correction.

The 16-bit DAC, D1, puts out an accurately set voltage equal to the desired output of D2, which consists of a two 8-bit DACs. D2A, with no attenuation, receives the eight most-significant bits. D2B's output is attenuated by a factor equivalent to 6 bits, and offset, so that it can add up to ± 2 LSBs to the output of D2A. The outputs of the two DACs are summed and compared with the output of D1. D2B is driven by a an up-down counter whose direction, determined by the sign of the comparator output, tends to drive the error towards zero. When the comparator changes sign, the count stops, and D2's output has been set to an accuracy of better than 14 bits, regardless of the actual setting of the digital input to D2B. If the system has a number of such DACs, each is set, in turn.

This arrangement tends to be somewhat slow, especially for large step changes in output, it uses a fair amount of hardware, and it doesn't make best use of available software. An alternative method (Figure 3.13) uses a fast successive-approximation a/d converter to measure the output of a linear amplifying comparator (viz., instrumentation amplifier) which compares the output of the reference, DAC1, with the output of DAC2A (DAC2B output = 0). The

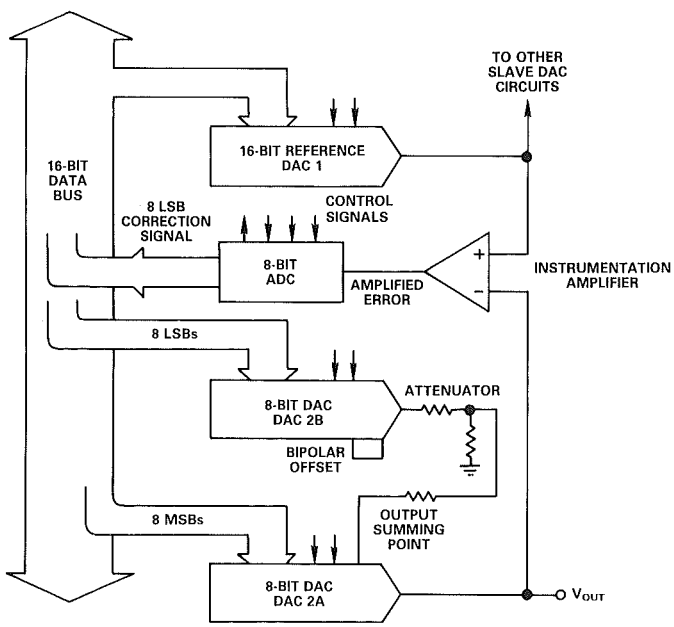


Figure 3.13. Master-slave scheme, using 8-bit ADC and software for correction.

digital output, which represents the difference of the analog values, is simply latched into the input of DAC2B. The total analog output of DAC2A and DAC2B is thus very nearly equal to that of DAC1, with a resolution of at least 14 bits; again, errors in DAC2A are taken into account in the setting of DAC2B.

For either of these schemes to be successful, the noise level and dc instability of DAC2A during the period between updates must be within a fraction of 1 LSB of the *overall* resolution.

3.9 ISOLATION

In any data-distribution system in which common-mode potentials pose a serious threat to the integrity of data, equipment, or organisms, it frequently becomes necessary to isolate the various analog loads from the digital data source. Otherwise, substantial differences in ground potential at the various locations could cause large ground currents, induced noise, or worse.

Isolation is accomplished by magnetic or optical coupling of either the digital or the analog signal. In one form of isolation, the analog output is isolated and then transmitted via a 4-to-20mA 2-wire current link, which is immune to voltage noise.

As a practical example of an isolated DAC, Figure 3.14 shows the architecture of a 10-bit systems DAC with 4-to-20 mA current output. The data, which might represent the setting of an actuator, is normally latched into the *preset* input of a latched 10-bit counter from an 8- or 16-bit data bus, in response to a WRITE command. The digital word is converted to an analog signal,

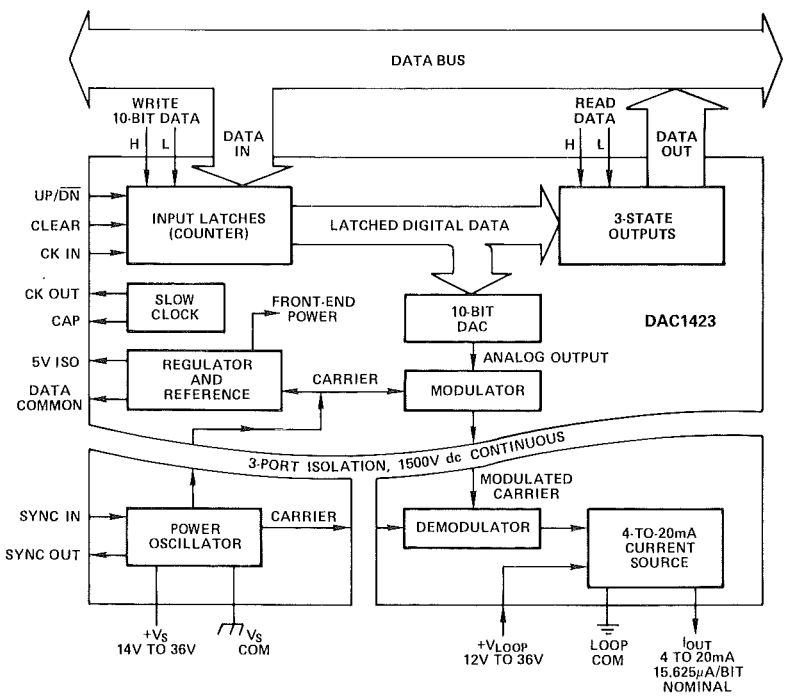


Figure 3.14. Block diagram of 10-bit System DAC having galvanic isolation, 4-to-20-mA output, data readback, power supply independent of bus system, and available external (e.g., manual) updating.

which is transmitted across an isolation barrier (via a modulated carrier), demodulated, and converted to a current signal with a 4-mA offset and a 16-mA span.

This DAC provides for preservation of the last setting and independent operation if—for any reason—useful data is not available from the bus. Under such conditions, pulses applied to the CLOCK input of the counter, while the UP/DOWN input is 1 or 0, can be used to increase or decrease the number stored in the register (and hence the analog output), one bit at a time. The pulses may be applied by manual switching, by an external backup processor, or gated in from an on-board slow clock generator.

As the Figure shows, there is also a three-state data-output register, which can place the information stored in the latches on the bus in response to a READ signal. This facility, which makes the DAC look like READ/WRITE memory, permits the computer to learn the state of the device (and the parameter it actuates) at any time—and especially when computer control is restored after an off-the-bus period (making possible "bumpless transfer" of control when the computer returns on-line). A CLEAR input can be used to set the latches to zero, e.g., during startup.

One of its most important attributes, the DAC has three divisions ("ports")—galvanically isolated from one another—with a breakdown rating of 1500V dc continuously or 1000V rms ac at line frequency for one minute. Its three sections might be termed: power, output, and front end. The *power* section contains a synchronizable high-frequency oscillator, the output of which is transformer-coupled to the other sections. The *front end* has a regulator and reference, a CMOS d/a converter, a modulator with transformer-coupled output, the slow clock for off-bus applications, and the digital logic circuitry. A small amount of isolated power is also available for external devices, such as logic gates for external drive in off-bus operation. The *output* section has the demodulator and current-output circuitry, with provisions for external offset and span adjustment. The current-loop power supply may be completely separate from the primary supply, or it may be the same supply.

Chapter Four

System Integration and Remote Data Acquisition

Now that we have seen a number of basic data-acquisition and data-distribution architectures in Chapters 2 and 3, it may be worthwhile to pause and consider some forms of system implementation. Areas of contemporary interest include:

1. Proprietary systems, subsystems, and components for interfacing and data communication.

2. Interfacing converters with nearby destinations, such as a microprocessor data bus, using parallel and byte-serial connections.

3. Using serial techniques to communicate sensor-based data with computers or distant destinations.

In Chapter 2, many of the configurations treat parallel data from the converter as an input to a nebulous "buffer" block. This buffer translates the converter's output into a machine data format, monitors the status of conversion, initiates conversions, addresses the multiplexer, controls sample-hold and gain-ranging, etc. These functions are achieved by purchasing proprietary interface products—ranging from ICs and modules to complete systems—and integrating them into the user's overall system.

In this chapter, we shall examine ways in which these forms of interfacing are achieved—first by outlining a number of proprietary systems and subsystems that form a hierarchy, then by a more-detailed consideration of interfacing techniques employing parallel and serial digital approaches, and the MACSYM ADIO (Analog-Digital Input-Output) bus.

As the preceding sentence may imply, we must be careful to limit the scope of this discussion, because any of these topics could itself justify a volume the

size of this book for a thorough in-depth treatment of all possible cases. Our method will employ the following approaches: First, this book is oriented primarily towards data acquisition and conversion; we shall seek to maintain that focus in this chapter. For that reason, (and because they are widely documented elsewhere), we will avoid treatment of the popular IEEE-488 instrumentation bus and CAMAC concepts. Second, though we will tend to summarize in somewhat general terms, we will limit much of our discussion to ideas for which concrete embodiments can be found in the Analog Devices product lines. This, in turn, permits an abbreviated overview, with security in the knowledge that the reader who desires greater depth can find substantial amounts of detailed information about specific approaches (and products available to implement them) in our published literature.

4.1 SUBSYSTEMS FOR INTERFACING CONVERTERS TO THE ANALOG AND DIGITAL WORLDS

Briefly summarizing some salient ideas from earlier chapters, data acquisition is the process of transforming electrical voltages or currents, usually transducer outputs, into digital information to be received at some defined destination in a system, for storage, display, processing, or further transmission (Figure 4.1). The data-acquisition process typically involves these forms of activity:

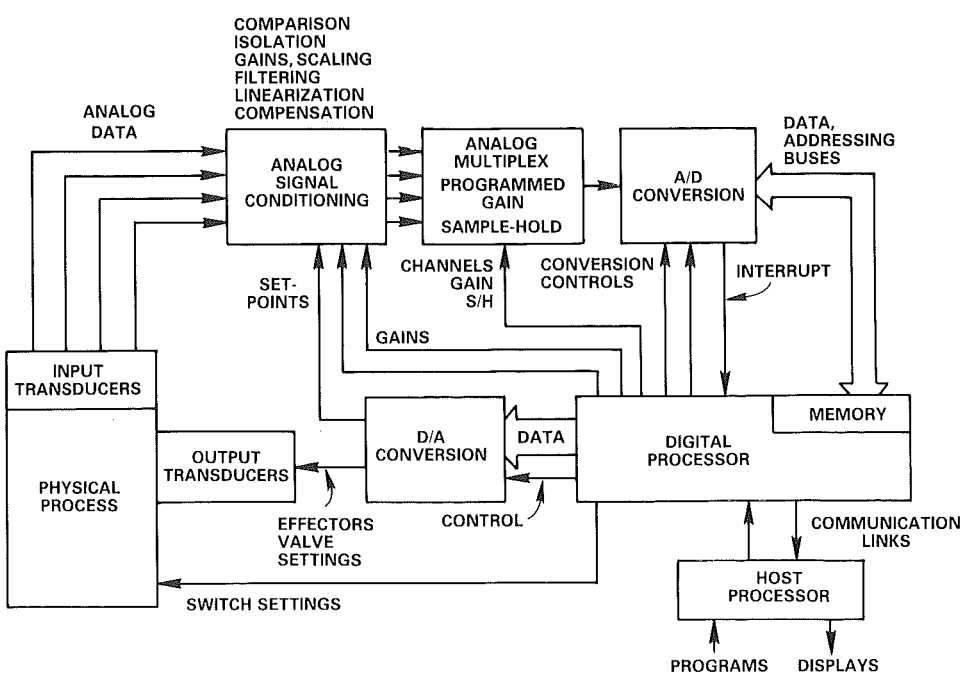


Figure 4.1. Data-acquisition function.

Analog signal-manipulation
Analog-digital conversion
Digital signal manipulation
Digital control manipulation, employing hardware and software

Analog signal manipulation includes such signal-conditioning operations as isolated preamplification, gain adjustment, linearization, algebraic functions (perhaps involving other inputs), sample/track-hold, and analog multiplexing.

An *analog-to-digital converter* produces a parallel or serial digital code that represents the ratio of an analog signal to a reference voltage or current. The digital code is usually—but not always—a binary or binary-coded-decimal number proportional to the ratio. Though widely and beneficially used, but not explicitly expanded upon here, a 1-bit a/d conversion can be the result of a go-no comparison of an analog signal with a set-point.

Digital SIGNAL manipulation might involve multiplexing, various arithmetic and logical operations—e.g., magnitude comparisons, algebraic operations, code or format conversions—storage, transmission to a slave or host processor, and deriving control signals for either digital handshaking or for operations on the "real-world" portion of the system.

Digital CONTROL manipulation includes control of all digital operations, programming of the analog functions (switching gains, channels, or circuit configurations), initiation of conversions, etc., and all of the associated software.

A *data-acquisition system* (for the present purposes) is an operationally self-contained subsystem, consisting of the conversion function (which involves at least one a/d converter and may involve d/a converters) and some portion of both the analog and digital manipulation circuitry. This definition (obviously quite flexible) permits a full functional range from a simple a/d converter (with a given analog span and digital controls) to an "intelligent" multichannel measurement-and-control subsystem—and a physical gamut from an integrated-circuit chip to a rack (or a room) full of equipment.

The desired properties of a given subsystem (in relation to the system) are determined by the application; their choice is affected by such factors as resolution, noise levels, system size and complexity, delegation of system tasks, frequency and level of interactions, the physical environment, software availability and compatibility, and (of course) cost—of hardware, of software, of wire, and of system development, prototyping, and manufacture.

When choosing the approach to take in designing a system, one system designer's component or subassembly may be another designer's turnkey system. In general, the design problem involves a classical "make-or-buy?" dilemma. The designer will purchase available components or subsystems that reflect the level of integration that is a best compromise between out-of-pock-

et cost and the many costs—both overt and hidden—of expending design and manufacturing effort in technological areas that are peripheral to one's primary mission.

The hierarchy of systems integration is especially deep at Analog Devices, ranging as it does from integrated-circuit converter chips to general-purpose computer-based data-acquisition subsystems, complete computer-based automatic test systems for integrated circuits and intelligent machine-vision systems. As examples of the levels of the hierarchy that are relevant for readers of this book, Table 4.1 (overleaf) lists some specific products that were available in 1985.*

We will review briefly the functional repertoire of these products in relation to the conversion interface, starting with the simplest and working our way up the chain.

4.1.1 THE AD574A INTEGRATED-CIRCUIT A/D CONVERTER

One of the simplest data-acquisition structures having a controllable interface capability is a bus-compatible a/d converter. A good example is the AD574A, a complete monolithic 12-bit a/d converter, furnished in a 28-pin dual in-line package (DIP). As the block diagram shows (Figure 4.2), it is a successive-approximation type with an internal 10-V reference. It accepts analog inputs with ranges of -5V to $+5$V, -10V to $+10$V, 0 to $+10$V, and 0 to $+20$V.

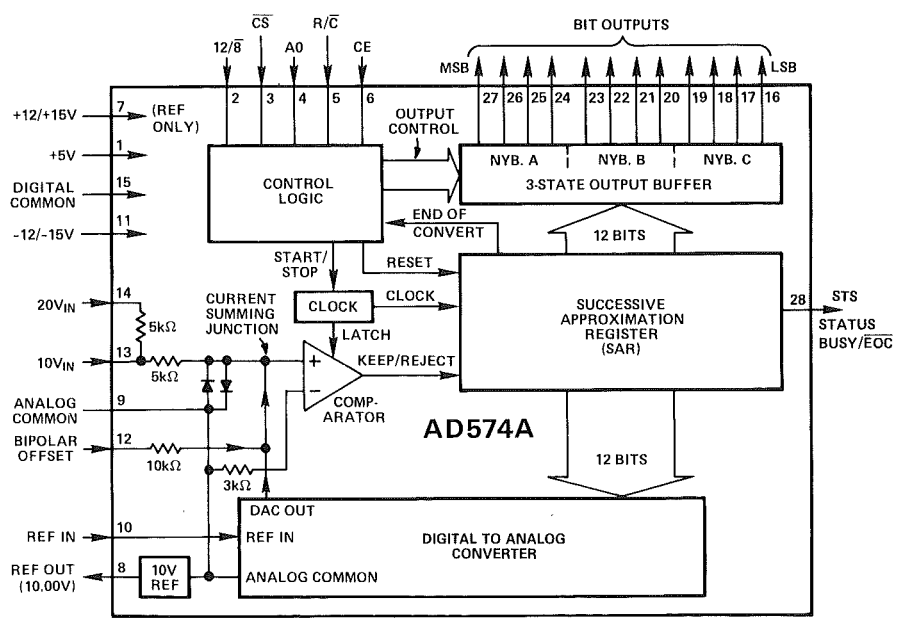


Figure 4.2. Functional diagram of 12-bit IC a/d converter.

*Since a complete discussion of their properties and applications is beyond the scope of this book, we will discuss several of these products in the context of their role in the conversion system. Complete information is available from the manufacturer, ranging from free data sheets and brochures to complete instruction books and software manuals at nominal cost.

Its digital output appears on three sets of three-state quad output latches and is left-justified. This means that the data represents the analog input as a fraction of full scale ranging from 0 to 4095/4096, implying a binary point to the left of the MSB (for example, .1100 0111 1001). The 12 bits of output data can be read either as one 12-bit word or as two 8-bit bytes—one with the 8 most-significant data bits (1100 0111), the other with 4 data bits and 4 trailing zeros (1001 0000). Conversions can be initiated either under program control, or in a stand-alone continuous-conversion mode. Its mode of interfacing will be discussed in Section 4.2.

4.1.2 THE AD364 DATA-ACQUISITION SYSTEM

The next level of simplification for the user (complexity for the device designer) adds a multi-channel analog front end. A graphic example of how this is done is the AD364, a complete microprocessor-compatible 12-bit data-acquisition system consisting of two complete functional blocks in hermetically sealed integrated-circuit packages (Figure 4.3). It essentially adds a multi-channel analog front end to an AD574A a/d converter.

The analog input section, in a 32-pin dual in-line package, consists of two 8-channel multiplexers, a unity-gain differential buffer amplifier with high input impedance, a sample(track)-hold, channel-address latches, and control logic. The multiplexers can be connected to the subtractor in either an 8-channel differential or 16-channel single-ended configuration, under the control of a logic-operated mode switch. This means that the AD364 can perform in either mode without external hard-wired interconnections. Of perhaps

(continued on page 74)

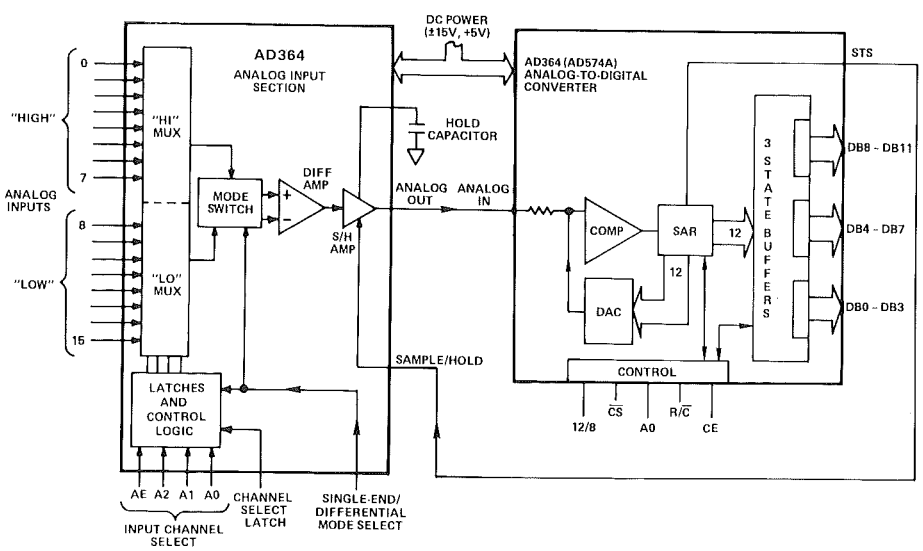
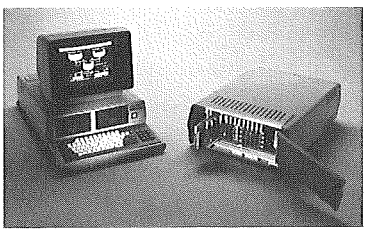


Figure 4.3. A/D converter and multichannel analog front end.

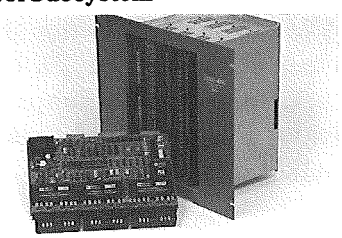
TABLE 4.1. REPRESENTATIVE EXAMPLES OF SYSTEMS, SUBSYSTEMS, AND SUBASSEMBLIES FOR DATA ACQUISITION MANUFACTURED BY ANALOG DEVICES IN 1985.

MACSYM 350 Fully Integrated Measurement And Control SYsteM



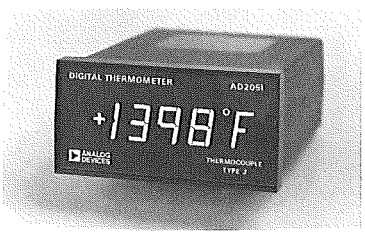
MACSYM 350 is a minicomputer-based measurement-and-control system used to automate the measurement, evaluation, and control of real-world phenomena, both analog and digital, while interfacing with human operators and other computers. It includes a keyboard, color display, and graphics, 5 ¼″ floppy-disk storage, and a data-acquisition subsystem. It can hold up to 16 interchangeable analog-digital input-output (ADIO) cards—which may be field-wired to sensors. Its powerful multitasking real-time language, Measurement And Control BASIC (MACBASIC) is quickly grasped by users without extensive prior programming experience.

μMAC-5000 Programmable Measurement-and-Control Subsystem



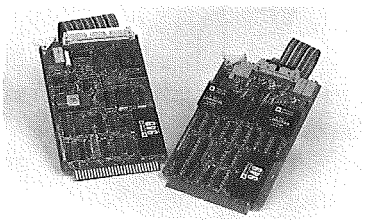
μMAC-5000 is a single-board programmable measurement-and-control system. Combining direct connection to sensors—via screw terminals—with modular signal conditioning, conversion, digital inputs and outputs, a 16-bit microcomputer, an extended BASIC language for measurement and control, serial communication facilities, a power supply with uninterruptible features, and ruggedized construction, it provides a compact, easily expandable instrumentality for measurement and control, applicable in a broad range of stand-alone or distributed control systems. It is also physically and electrically compatible, serving the function of host, with μMAC-4000 subsystems, which have similar front ends, but limited computation and communication capability (Section 4.3.3).

AD2051 Microprocessor-Based Thermocouple Meter



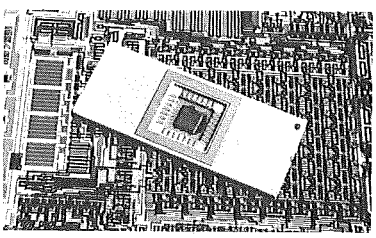
The AD2051 connects directly to a switch-selected thermocouple (J, K, T, E, R, or S), corrects for the cold-junction temperature, amplifies and linearizes the thermocouple output, calibrates itself, provides a display in degrees Celsius (−165°C to +1760°C) or Fahrenheit (−265°F to +1999°F), and makes available a digital output, in the form of 7-bit character-serial ASCII. Also optionally available are a linearized analog output and facilities for a full-duplex 20mA isolated digital loop for communication with a computer or terminal.

RTI-1260 and RTI-1262 Microcomputer Analog I & O Subsystems



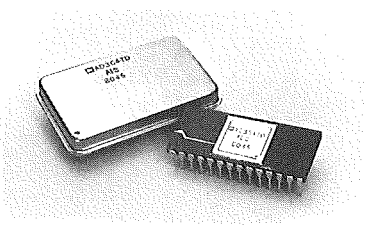
The RTI-1260 and RTI-1262 series of "Real-Time Interface" Analog I/O Subsystem Cards provide an analog input/output facility for microcomputer systems employing the popular STD bus. Interfacing as a block of memory locations, they fit easily into programs for microprocessors such as 8080A, 8085, 6800, 6809, and Z80. Capable of multiplexing 16 differential or 32 single-ended input channels, the RTI-1260 has a programmable-gain amplifier, sample-hold, and 12-bit a/d converter; the RTI-1262 has four channels of 12-bit d/a conversion. These cards interface with the STD Bus, but RTI-series Input, Output, and combined Input/Output boards, and others like them, are available for popular microcomputer bus structures, including the Multibus, TM990 Bus, LSI Bus, VME Bus—and as plug ins for PC slots.

AD7581 μP-Compatible 8-Bit 8-Channel Memory DAS



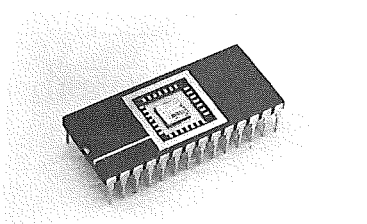
The single-chip CMOS AD7581 continuously scans 8 analog input channels, converts them to digital, and stores the data in bus-addressable RAM (read-write memory). The AD7581 interfaces directly with 8080, 8048, 8085, Z80, 6800, and other microprocessor systems. Data can be read at any time for any channel; on-chip logic provides interleaved direct memory access (DMA).

AD364 16-Channel 12-Bit Data-Acquisition System



The AD364 is a complete data acquisition system in the form of two integrated-circuit packages. It includes a versatile multiplexer, differential amplifier, sample-hold, and 12-bit a/d converter. The multiplexer, and its associated mode switch, will handle from 8 differential to 16 single-ended channels, as well as intermediate combinations. The converter's 12-bit output will interface directly with 8- or 16-bit microprocessor buses, under software control.

AD574A μP-Compatible Analog-to-Digital Converter



The AD574A is an integrated-circuit 12-bit a/d converter. Accepting a variety of analog input voltage ranges, it interfaces to most popular microprocessor types having an 8-, 12-, or 16-bit data bus—without external buffers or peripheral interface controllers.

(continued from page 71)

greater significance, one AD364 can serve a mixture of both single-ended and differential sources under software control.

Multiplexer channel-address inputs are interfaced through a level-triggered (transparent) input register. With logic 1 at the Channel-Select Latch, the address signals feed through the register to directly select the appropriate input channel. This address information is latched into the register on the transition from logic 1 to logic 0 at the Channel-Select Latch input. The latching feature is useful when the user has no control over when input channel-address information may change—for example, when it is provided from an address, data, or control bus that may be required to serve many devices. Internal logic monitors the status of the differential/single-ended mode input and addresses the multiplexers according to an established scheme.

The sample-hold mode control input is normally connected to the status output from the a/d converter. When a conversion is initiated by applying a Convert Start control sequence (see AD574 Interfacing in Section 4.2), the Status goes High, putting the sample-hold into the Hold mode, freezing the information to be digitized for the period of conversion. When the conversion is complete, Status returns to logic 0 and the sample-hold tracks the input until the next conversion is initiated.

4.1.3 AD7581 8-BIT 8-CHANNEL MEMORY DAS

A quite sophisticated system capability on a chip is exhibited by the AD7581, a complete 8-channel, 8-bit data-acquisition system on a single monolithic chip. It consists of (Figure 4.4) an 8-bit ratiometric successive-approximation

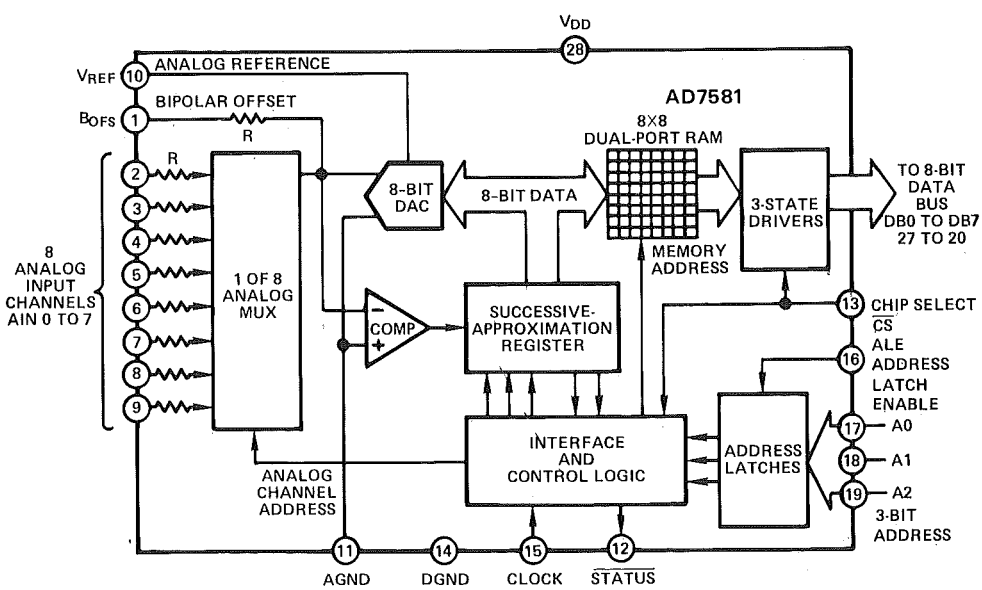


Figure 4.4. 8-channel, 8-bit data-acquisition system with memory.

a/d converter, an 8-channel analog multiplexer, an 8 × 8 dual-port random-access memory (RAM), three-state drivers (for interface), address latches, and microprocessor-compatible control logic. When used with appropriate references, it accepts either unipolar inputs (0 to +10V) or bipolar inputs (−5V to +5V, offset-binary output). It converts each channel in turn and stores the output; the digital value corresponding to any channel's input can be read from the AD7581's memory at any time. Its timing and operation with a microprocessor bus will be discussed in Section 4.2.

4.1.4 RTI-1260 AND RTI-1262 MICROCOMPUTER ANALOG INPUT AND OUTPUT SUBSYSTEMS

At the next level of complexity are "real-time interfaces" to standard microcomputer and personal-computer buses. When the intended function of a microcomputer system is measurement and control of real-world (i.e., analog) phenomena, analog I/O cards, which contain data-conversion components and bus interface logic, are necessary to interface the computer with the real world. They are designed to relieve the system designer of physical and electrical hardware problems relating to the analog-digital-processor interface (and many of the software considerations, too). RTI families, including input-only, output-only, and input/output boards, are available for popular buses, such as MULTIBUS, STD Bus and LSI-11 Bus.

The RTI-1260 Analog Input Subsystem and the RTI-1262 Analog Output Subsystem are representative of interface cards designed to work directly with

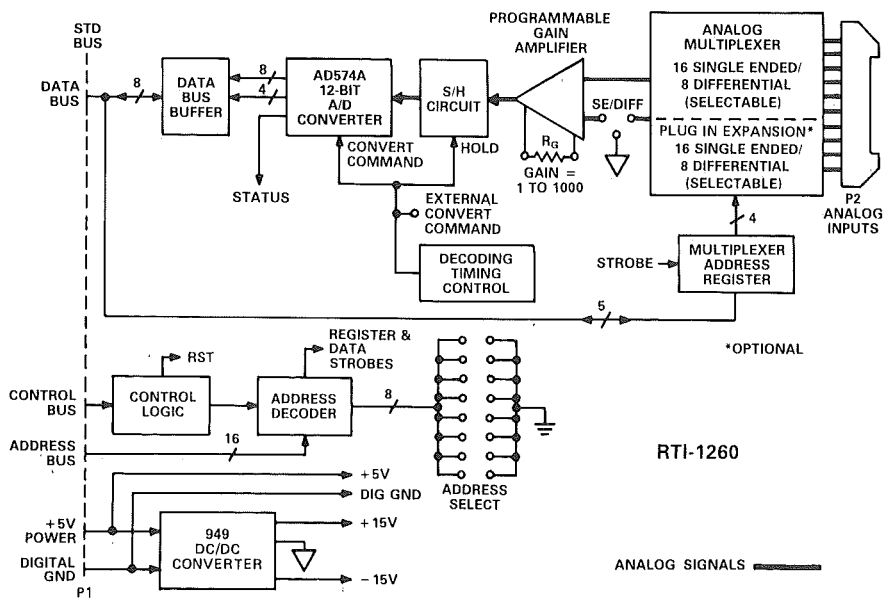


Figure 4.5. Functional diagram of data-acquisition card for STD bus.

microcomputer bus structures, interfacing in the same way as memory. They are designed for complete physical, electrical, and software compatibility with the STD bus, a popular bus that is compatible with many microprocessor types.

Figures 4.5 and 4.6 show the salient electrical features of the RTI-1260 Analog Input Card and the RTI-1262 Analog Output Card. The upper portions of the diagrams show conventional data-acquisition and distribution architectures. The lower portions detail the direct connections to the microcomputer buses: data, control, address, and power. A dc-dc converter converts the +5V bus supply voltage to the low-noise, isolated ±15V required for the analog circuitry.

Input cards for data acquisition. The basic architecture of an analog input card is shown in Figure 4.5. Analog inputs arrive at the terminals of a multiplexer, which selects one of 16 or 32 channels. The multiplexer can be configured for single-ended, differential, or quasi-differential modes. The single-ended mode is used when all signals are referred to a common ground—and are of sufficient magnitude in relation to noise to provide appropriate resolution; 16 or 32 channels are optionally available. For noisier environments, or where signals come from sources at differing common-mode levels, the differential mode can be used by the pairing of signal inputs to minimize the effects of common-mode noise; this halves the number of available inputs. If all signals have a common connection (not at system ground), it can be used as one side of the differential input; this quasi-differential connection takes advantage of the amplifier's differential inputs without sacrificing channel capacity.

The outputs of the multiplexers (which use dielectric isolation and can handle signals of up to ±35V without damage) feed a differential-input instrumentation amplifier, having gains programmable from 1 to 1000, to amplify the signal (±10mV to ±10V full scale) to the specified input range of the converter. The sample-hold tracks the signal and freezes it during a/d conversion. The converter produces an 8-, 10-, or 12-bit digital representation of the signal, and this result is made available to the microcomputer bus, via a set of three-state program-controlled registers.

The multiplexers accept software-determined commands from the microcomputer to select a specific analog channel and start an a/d conversion.

Output cards for control. Analog output cards (Figure 4.6) contain independent d/a conversion channels for driving chart recorders, servomechanisms, control valves, and output transducers. The analog output is set by writing a digital code to the appropriate address. If the resolution of the data exceeds 8 bits, the DAC requires two bytes of data. The DACs are double-buffered, so that both bytes may be separately loaded into the input register and then strobed simultaneously into the DAC, avoiding intermediate outputs and insuring cleaner transitions from one output value to the next.

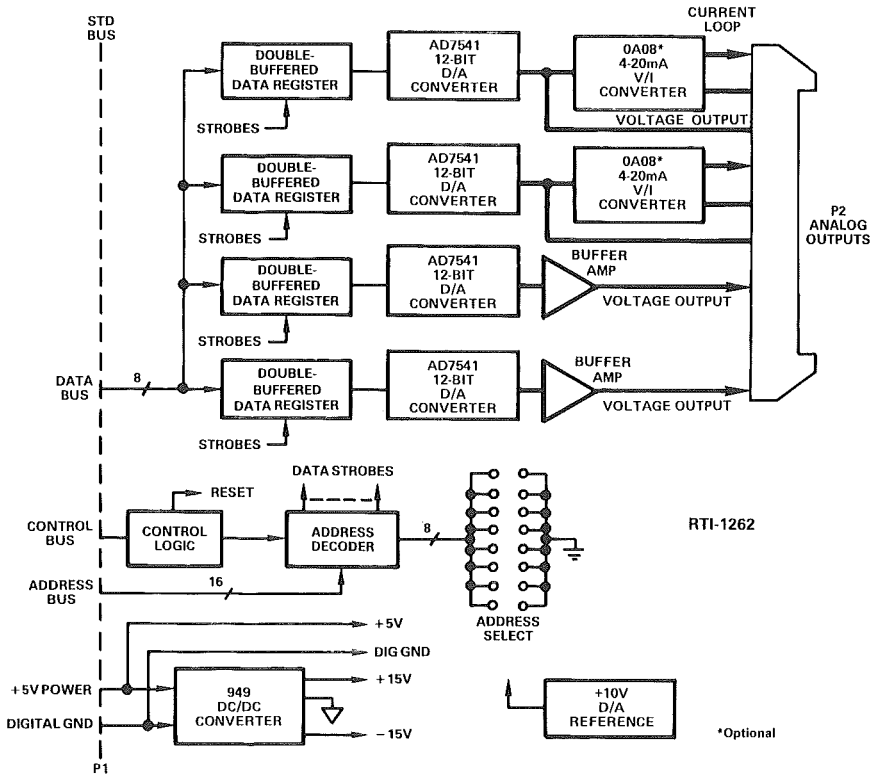


Figure 4.6. Analog output card for STD bus.

Usually, the cards offer single-ended or bipolar voltage output at standard levels. However, in some applications (e.g., automatic control), where the load may be at some distance from the microcomputer system, it is useful to transmit the analog signal as a current, rather than as a voltage, for immunity from voltage noise and IR drops in the output wiring. Current-output options typically provide an output current span from 4mA to 20mA. Digital interfacing of the RTI devices will be discussed in Section 4.2 of this chapter.

4.1.5 AD2051 μP-BASED THERMOCOUPLE METER

So far, we've discussed devices and subsystems that interface with high-speed multiwire microprocessor buses. Since long runs of parallel bus wire are high in noise, crosstalk, capacitance, and cost, it is usually imperative that the converter and its associated circuits be nearby—usually in the same card cage. For many applications, however, this may require that analog signals be carried over lengthy wire runs, with excellent prospects for signal degradation. The subsystems to be considered below can be operated closer to the signal source; they can communicate with computers by the use of serial transmission, and— because they have built-in processing capability and memory— they can stand alone in their transactions with analog signal sources.

The AD2051, for example (Figure 4.7), is a digital panel instrument that connects directly to a thermocouple. Switch-selected for the correct thermocouple type (J, K, T, E, R, or S), it corrects for the cold-junction temperature, amplifies and linearizes the thermocouple output, calibrates itself, provides a display in degrees Celsius (−165°C to +1760°C) or Fahrenheit (−265°F to +1999°F), and makes available a digital output, in the form of character-serial (7-bit parallel) ASCII. Also optionally available are a precision analog output and facilities for a full-duplex 20mA isolated digital loop for communication with a computer or terminal. These terms, and the ways in which the AD2051 interfaces, will be discussed in part 4.3.

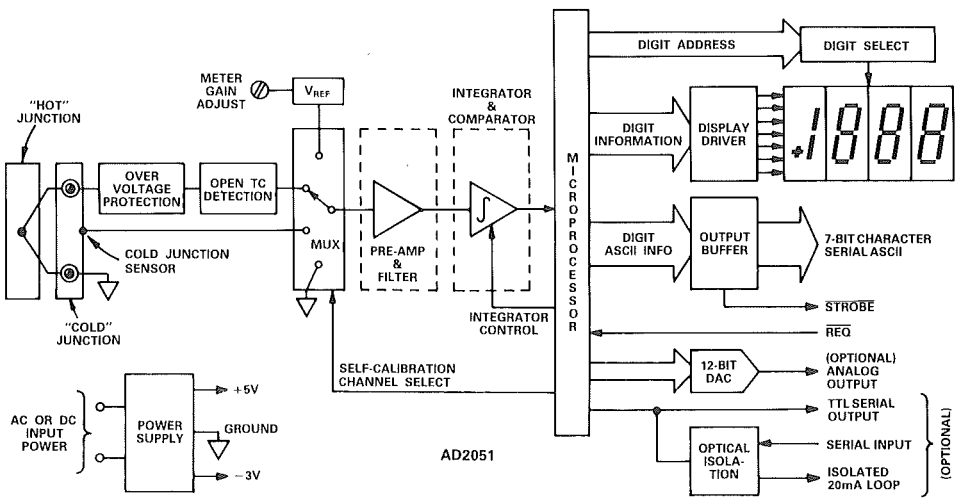


Figure 4.7. Versatile thermocouple thermometer.

4.1.6 μMAC-5000 PROGRAMMABLE SINGLE-BOARD MEASUREMENT-AND-CONTROL SYSTEM

The μMAC-5000 combines, on a single 9 ½″ × 13″ (241.3 mm × 330.2 mm) board, signal conditioning, multiplexing, and a/d conversion for 12 channels of analog sensor input (thermocouples, strain gages, etc.) It also has 8 digital outputs and 8 digital inputs, for communicating logic states, switch closures, etc. Additional channels are available via an expansion port and a family of analog and digital expansion boards that can be housed in the same card cage.

Fully programmable, employing μMACBASIC, a powerful, easy-to-use programming language, its 5-MHz, 16-bit 8088 on-board CPU processes the data collected from the input channels, sends out control signals, and comunicates with other equipment as directed by the program installed in its memory. Since the memory available over and above that required for the μMACBASIC operating system and internal operation is at least 16K bytes of both RAM and ROM, the user has the option to download programs into read-write memory from a host computer or to store them in non-volatile EPROM.

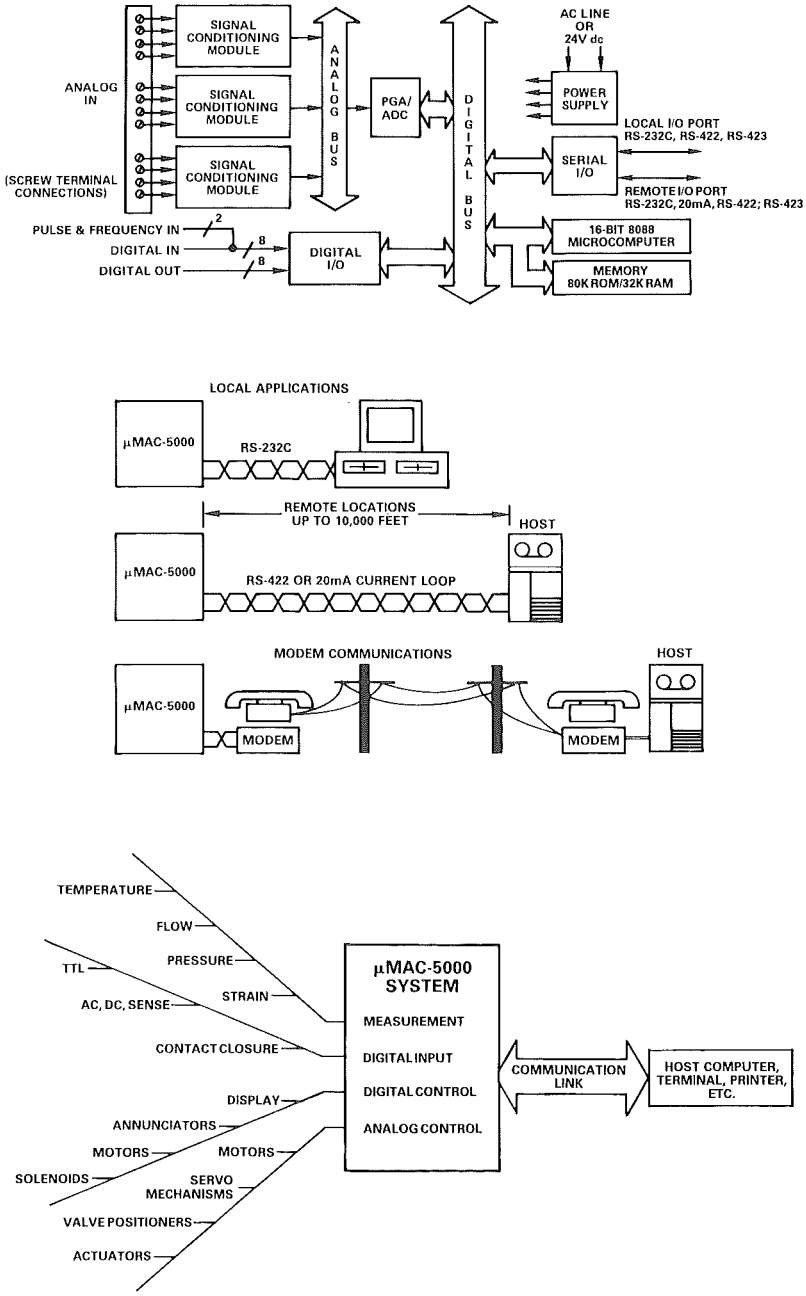


Figure 4.8. Functional diagram, typical communication modes, and system orientation for single-board measurement-and-control system.

The μMAC-5000 can be used as a stand-alone measurement-and-control system; it can provide remote intelligence for servicing a host computer; or it can act as a host for a μMAC-4000 slave measurement-and-control subsystem (see

Section 4.3.3). Figure 4.8 shows the basic structure of the μMAC-5000, its role in a system, and some of the ways it can communicate locally with terminals and printers—and either remotely or locally with host computers and other entities.

Analog signals arrive at the inputs of three plug-in quad signal-conditioner/multiplexers, which can be chosen from a variety of available types to provide such functions as cold-junction compensation for thermocouples, adjustable-gain preamplification, high-voltage isolation, and sensor fault indication. The function of each individual input is selectable in software. Each module's output is multiplexed onto the internal analog bus; the resulting signal is amplified by the programmable-gain amplifier (PGA) to fit the input range of the a/d converter, and the result of the conversion— appearing on the digital bus—is processed as required by the program. An integrating converter provides resolutions from 13-bits-plus-sign to 11 bits, depending on the desired number of conversions per second.

The μMAC-5000 software is easy to use. For example, on the command 'AIN(channel)', it automatically addresses the voltage on the specified analog input channel and converts it to digital; if identified as originating in a thermocouple, the data is linearized, compensated, and translated into engineering units. Programmers are relieved of writing these steps into their program. The use of an analog-input command AIN illustrates this:

```
10 TEMP1 = AIN(1)
20 PRINT "THE TEMPERATURE IS ";TEMP1;" DEGREES C."
```

When line 10 is executed, if the addressed, converted, and processed value of TEMP1 is 23.2, the displayed or printed response in line 20 would be:

```
THE TEMPERATURE IS 23.2 DEGREES C.
```

The first command measures the analog input on channel 1 (e.g., a thermocouple), computes the correct temperature in degrees Celsius, and assigns the value to the variable TEMP1. The second command formats and prints the data on a CRT or printer.

Besides such simplifications as these, μMACBASIC can further simplify the writing of programs by allowing the programmer to name Procedures or Functions. For example, in

```
10 IF AVG(3,10) > 700 THEN ALARM (2)
```

When line 10 is executed, a Procedure, AVG(3,10), is called (e.g., the averaging of 10 readings of analog channel 3), the result is compared with 700, and if the result is greater than 700, an alarm function (perhaps the outputting of a switch closure) is performed. The Procedure and Function may be written with any desired local line numbers and variable names without conflicting with the main program.

In Section 4.3.3, we will discuss the means by which the μMAC-5000 and other entities, such as personal computers, communicate with the response-only "intelligent" μMAC-4000 Measurement-and-Control Subsystem.

4.1.7 MACSYM 350 COMPUTER-BASED MEASUREMENT-AND-CONTROL SYSTEM

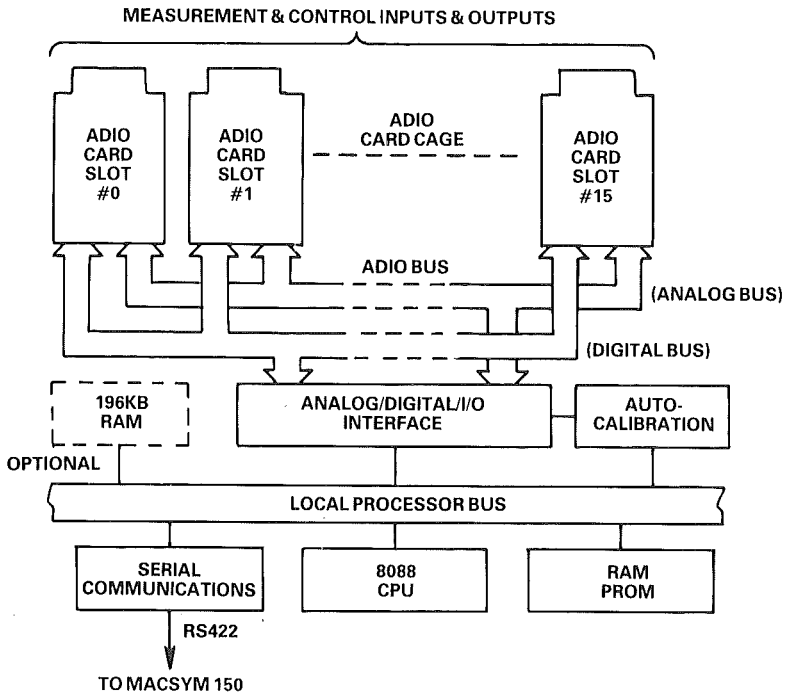
At the highest level of data-acquisition system integration, MACSYM 350 is a fully integrated Measurement And Control SYsteM developed specifically to acquire, reduce, store, present, and output real-time information in laboratory, process control, and discrete manufacturing applications. From architecture and packaging to software and documentation, the system is human engineered to minimize the time and experience required to configure, hook up, program, and operate in the user's environment.

The complete MACSYM 350 system, with integral signal conditioning, is packaged in two compact desktop units: the stand-alone MACSYM 150 workstation and the MACSYM 200 intelligent front end. The basic system, the MACSYM 150, includes a high-speed 16-bit 8086 processor and 8087 numeric coprocessor, keyboard, color display, and 5 ¼" floppy-disk storage. Specialized I/O cards plug into a 6-slot internal backplane to provide analog, digital and communications input/output, and memory expansion.

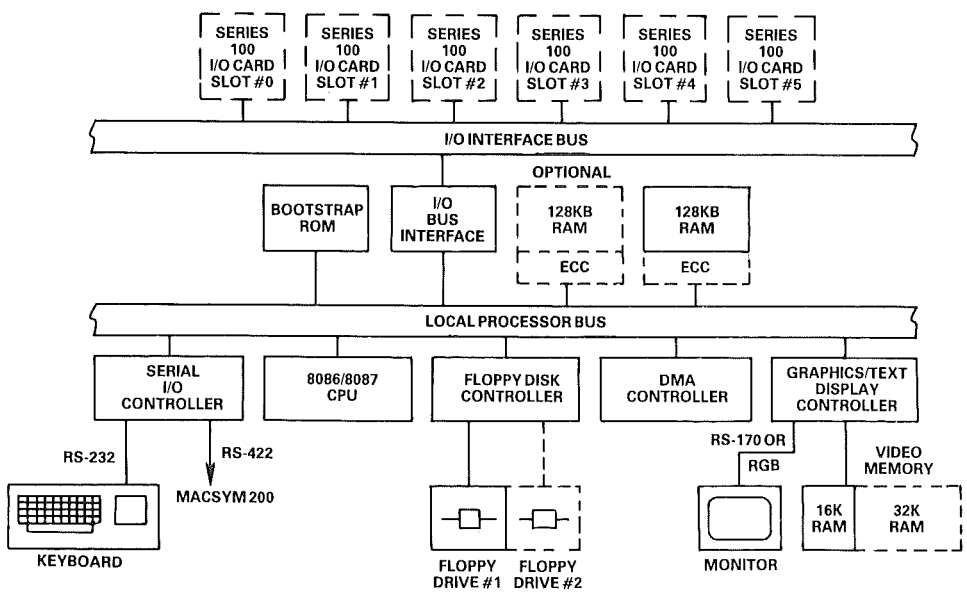
It can be expanded to form MACSYM 350 systems, systems of much larger scope and range of functions, by the addition of one or more MACSYM 200 intelligent front ends, each of which provides for up to 16 additional slots for data-acquisition and control. A large library of analog/digital input/output (ADIO) cards—which can be interchangeably plugged into any of 16 card slots provided in the MACSYM 200's chassis—is available. Application programs can be written immediately in MACBASIC, a multitasking real-time BASIC, optimized specifically for measurement and control.

Figure 4.9 is a simplified overall block diagram of MACSYM 150 and 200, depicting the organization of the major components comprising such systems. The 16-bit central-processing unit has hardware floating-point and a minimum of 128K bytes of RAM. The system-control card includes a 24-hour real-time clock and console serial interface with RS-232C or 20-mA current loop (110 to 9600 baud).

A key element of MACSYM systems is the dual bus system. The processor communicates with memory and computer-type peripherals via the conventional computer bus structure. However, communication with the analog and digital I/O cards is established by way of the ADIO controller, an intelligent interface which provides a number of shared functions—including a/d conversion. The dual bus structure and shared functions minimize the complexity and cost of the individual cards and isolate the I/O bus from the noisy high-speed processor bus, permitting improved performance to be obtained with low-level analog signals. Since the ADIO controller deals with each card on



a. MACSYM 150.



b. MACSYM 200.

Figure 4.9. Simplified block diagram of minicomputer-based measurement-and-control system.

the basis of its own identity, it is possible to utilize any assortment of cards, 16 per MACSYM 200 chassis.

Capabilities of the signal-conditioning card family include low-level analog in/analog out, direct sensor interfaces (thermocouple, strain gage, RTD, etc.), digital input/output, isolation, and a wide variety of special functions. Optional screw-terminal boards permit direct connections to sensor wires. The number of cards used in a a given application may be greatly increased by the use of extension chassis. A variety of peripherals, from disk drives and plotters to graphic terminals and air-conditioned NEMA cabinets are available.

MACSYM 350 can interface with the IEEE-488 (1978) general-purpose instrumentation bus (GPIB); it interfaces via the ACP100 I/O card and its associated software driver, which enables MACSYM 350 to send and receive data and operate as a bus controller or listener/talker, using BASIC, for supporting up to 15 external devices simultaneously, with 9 operational functions and 30 available programmable instrument addresses, at a 2000-byte-per-second maximum data rate.

A/D conversion in MACSYM 350 systems is achieved in software by the variable, AIN(card slot, channel); d/a conversion is effected by AOT(card slot, channel). Thus, the powerful MACBASIC statement,

$$\text{AOT}(8,3) = 5*\text{AIN}(2,5) + \text{AIN}(3,6) + \text{K}(5) \tag{4.1}$$

means that when the statement is executed, the voltage applied to channel 5 of the card in slot#2 is measured, multiplied by 5, added to the voltage measured at channel 6 of the card in slot #3, added to a variable identified as K(5), and that sum updates the analog output from channel 3 of the card in slot #8, all within milliseconds.

MACSYM 350 has been described here to illustrate the highest level of integration of conversion into a general-purpose data-acquisition system for measurement and control. Designed for end-user convenience, MACSYM's a/d and d/a conversion functions are internal, fully integrated, and buffered from the world by the ADIO bus and software. Because MACSYM functions essentially as a self-contained turnkey system, supported by a panoply of tutorial publications, instruction and software manuals, application briefs—and a field service organization—the details of its converter-interfacing methodology are beyond the scope of these pages.

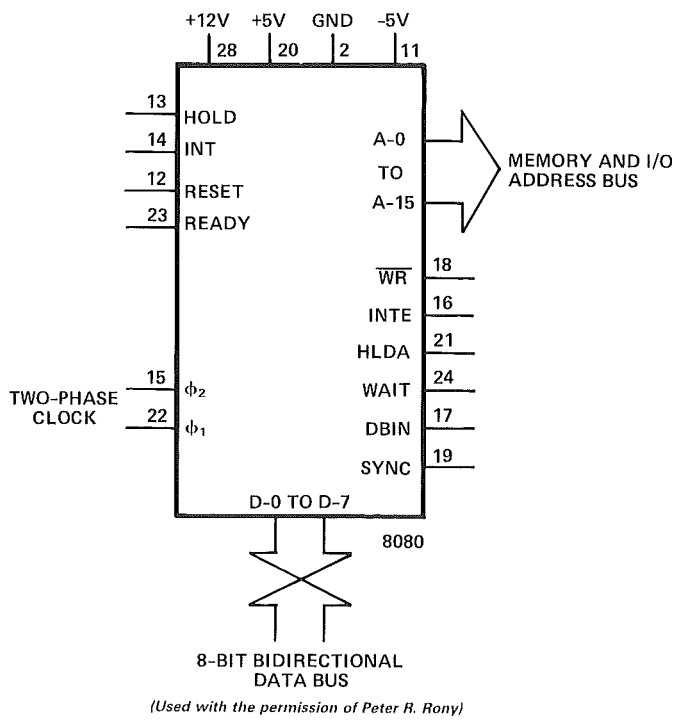
4.2 INTERFACING CONVERTERS WITH MICROPROCESSORS, USING PARALLEL CONNECTIONS

Microprocessors, because of their low cost and ready availability, have become established as the prime arena for computer interfacing. It therefore makes sense to indicate how the general principles discussed in earlier chapters can be applied in fairly specific ways to microprocessors.

There is a plethora of detailed microprocessor architectures and software systems, differentiated by manufacturer, by generation, and by degree of integration. It would be futile and well beyond the scope of this chapter to explore even a few of them in detail. Instead, we shall seek to show the elements that most microprocessors have in common and to indicate how some of the conversion and data-acquisition devices mentioned earlier can be interfaced to μPs. Many of the underlying ideas can be found in several of the references (at the back of the book), as well as in manufacturers' literature.

A *microcomputer* is an operational computer system, with a specified amount of memory, based on a microprocessor (CPU) chip. A *microprocessor chip* (packaged integrated circuit) is something less than a microcomputer, and the difference between the two is simply a measure of a continually shrinking technological gap.

Figure 4.10 shows a functional diagram of the connections to an 8080 microprocessor. They include a 16-bit unidirectional latched address bus, which is used to address one 8-bit byte out of a possible 65,536 bytes (64K) of external memory; an 8-bit bidirectional data bus for transferring data to or from the processor; a set of power-supply terminals; a pair of clock terminals; and a set of incoming and outgoing control lines. The processor itself contains an accumulator, a set of registers, and the operational capability of carrying out up to 256 different instructions, coded in 8-bit words. The instruction groups



(Used with the permission of Peter R. Rony)

Figure 4.10. Typical microprocessor interface connnections.

include data-transfer; arithmetic operations; logic operations; branching operations; and stack, I/O, and machine-control operations.

Double-precision operations are inherent: a number of instructions string two 8-bit data bytes together as a 16-bit word. It will be seen that this is a useful feature in dealing with converters.

4.2.1 MICROPROCESSOR INTERFACING, I/O vs. MEMORY

To interface a converter or a data-acquisition system to a microprocessor, a number of requirements must be fulfilled:

It must be possible to address the converter subsystem, and if a MUX with random addressing is used, it must be possible to address specific analog channels.

The output of the converter must be transformed to a compatible format and to circuitry compatible with three-state busing.

Suitable software and control signals must be provided to initiate conversion, determine when conversion is complete, and transfer the data appropriately.

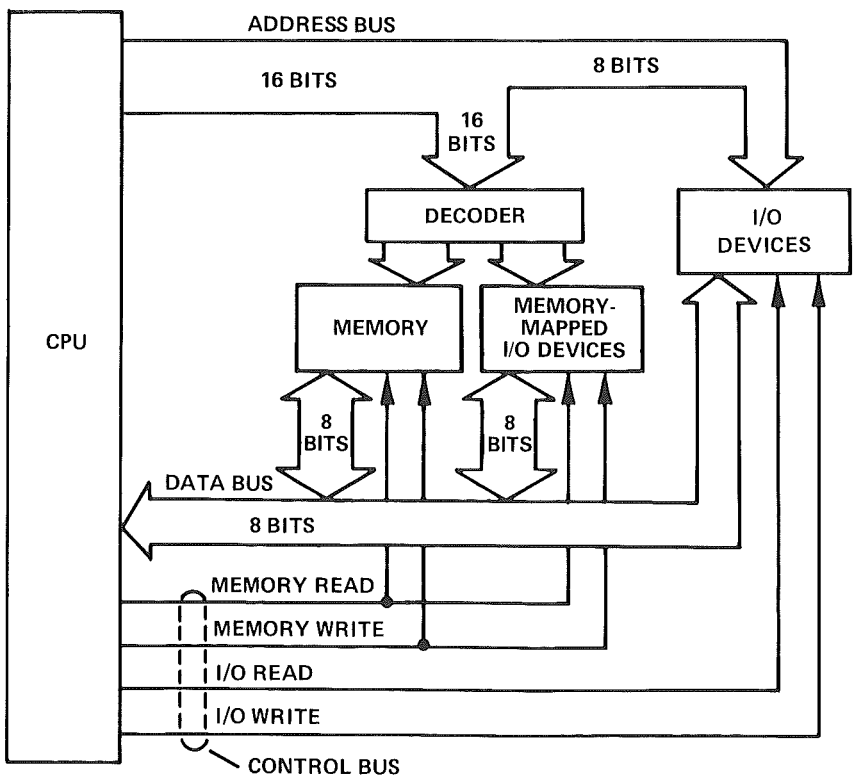


Figure 4.11. Accumulator (or isolated) I/O vs. memory-mapped I/O.

Analog-to-digital converters, like other I/O devices, may be interfaced tomicroprocessors by several methods. These methods include (but are not limited to) direct memory access (DMA), isolated, or accumulator I/O, and memory-managed (or memory-mapped) I/O. Direct memory access is the fastest, since conversions occur automatically, and data updates into memory are transparent to the processor. DMA logic is very processor-dependent and makes use of dedicated specialized hardware. Memory-managed and accumulator I/O are more-often used and somewhat easier to understand (Figure 4.11).

Memory-managed I/O assigns the I/O device to one or more locations in the memory space of the microprocessor. This technique has the advantage that the full range of memory-reference instructions may be used to operate on the data. The potential disadvantages include limiting the memory space available for program and data memory, somewhat more complex address decoding, and increased difficulty of isolating device-select pulses for system debugging. Many processors offer only memory-mapped I/O.

Accumulator I/O uses a set of control signals which are distinct and different from the memory control signals. These control signals, combined with use of a portion of the address bus, serve to define a totally separate I/O address space. This architecture is simpler from the hardware standpoint, since address-decoding requirements are less severe for the smaller space (for example, 256 inputs and 256 outputs may be used with the 8080), and distinct I/O Read and Write pulses are more easily located for system debugging purposes. However, processors using accumulator I/O generally can only send data to an output device from the accumulator. This can make the software more cumbersome, since processor-controlled transfers of I/O device data to a memory location cannot be accomplished in a single instruction.

Concrete examples of some of these concepts can be given in terms of some of the devices mentioned in this book.

4.2.2 INTERFACING AN IC 12-BIT A/D CONVERTER WITH MICROPROCESSORS

AD574A Controls. The AD574A contains on-chip logic for initiating conversions and reading out data, using signals commonly available in microprocessor systems. Referring to Figure 4.2 and Table 4.2, three of the logic inputs, CE, $\overline{CS}$, and R/$\overline{C}$, control the operation of the converter; the register-control inputs, A0 and 12/$\overline{8}$, control converted word length and data format. The state of R/$\overline{C}$ (read/convert), when CE (chip enable) and $\overline{CS}$ (chip select) are both asserted, establishes whether a Data-Read (R/$\overline{C}$ = 1) or a Convert (R/$\overline{C}$ = 0) operation is in progress.

The A0 line is usuallytied to the least-significant bit of the address bus. If a conversion is started with A0 low, a full 12-bit conversion cycle is initiated;

CE	$\overline{CS}$	R/$\overline{C}$	12/$\overline{8}$	A0	Operation
0	X	X	X	X	None
X	1	X	X	X	None
1	0	0	X	0	Initiate 12-Bit Conversion
1	0	0	X	1	Initiate 8-Bit Conversion
1	0	1	+5V	X	Enable 12-Bit Parallel Output
1	0	1	Dig. Com.	0	Enable 8 Most Significant Bits
1	0	1	Dig. Com.	1	Enable 4LSBs + 4 Trailing Zeroes

Table 4.2. AD574A truth table.

if A0 is high during a Convert start, a shorter 8-bit conversion cycle results. During data Read operations, the state of A0 determines whether the 3-state buffers containing the 8MSBs of the conversion result (A0 = 0) or the 4 LSBs (A0 = 1) are enabled. The 12/$\overline{8}$ pin determines whether the data is to be organized as two 8-bit words (12/$\overline{8}$ wired to Digital Common) or a single 12-bit word (12/$\overline{8}$ wired to VLOGIC).

The functions of the controls are summarized in Table 4-2. An output signal, STS, indicates the status of conversion. STS goes high at the beginning of a conversion and returns low when the conversion cycle is complete.

Microprocessor Interfacing. A typical a/d converter interface routine involves several operations. First, a Write to the ADC address initiates a conversion. The processor must then wait for completion of the conversion cycle, since most integrated-circuit ICs take longer than one machine instruction cycle to complete a conversion. Valid data can only be read after the conversion is complete.

The AD574A's STS (Status) signal indicates when a conversion is in progress. The processor can poll the signal by reading it through an external three-state buffer (or other input port). The STS signal can also be used to generate an Interrupt when the conversion is completed, if the system timing requirements are critical, and if the processor must perform other tasks during the time required for conversion (35μs for the AD574A) and must use the results of conversion as soon as they are ready. Another useful time-out method is to assume that the ADC will take 35 microseconds to convert, and insert a sufficient number of instructions to ensure that at least 35μs of processor time are consumed (or rely on an independent counter or clock, freeing the processor).

Once it is established that the converter has finished its cycle, the data can be read. If the ADC has 8 bits or less of resolution, a single data Read operation is sufficient. In the case of converters with more data bits than are available on the bus, a multi-byte data format is required, and multiple Read operations are needed. The AD574A includes internal logic to permit direct interfacing to 8- or 16-bit data buses, selected by connecting the 12/$\overline{8}$ input

low or high. In 16-bit bus applications (12/$\overline{8}$ high), the data lines (DB11 through DB0) may be connected to either the twelve most-significant or 12 least-significant bits of the data bus. The remaining four bits should be masked in software. The effect of the former is a left-justified fractional-binary format (1111111111110000, while the effect of the latter is a right-justified integer-binary format (0000111111111111). The interface to an 8-bit data bus (12/$\overline{8}$ low) is always done in a left-justified format with the AD574A. When the LSB of the address word is even (A0 Low), the eight most-significant bits (DB11 through DB4) are addressed; when the address is odd (A0 High), the output to the bus consists of 4 LSBs (DB3 through DB0), followed by four trailing zeros—bit-masking is unnecessary (Figure 4.12).

	D7							D0
XXX0 (EVEN ADDR):	DB11 (MSB)	DB10	DB9	DB8	DB7	DB6	DB5	DB4
XXX1 (ODD ADDR):	DB3	DB2	DB1	DB0 (LSB)	0	0	0	0

Figure 4.12. Left-justified data format for interfacing a 12-bit converter to an 8-bit bus.

Access time and data latency time for the AD574A's 3-state buffers are comparable to those for today's memory devices. Therefore, the AD574A can interface directly to many processor buses without the need for Wait states or external data buffers. We will show an example here of interfacing to 6800/6502 microprocessor systems. Interfaces to other systems, including the Apple II computer, may be found in the AD574A data sheet.

6800 Interfacing. The control signals and bus architecture of the 6800 and 6502 series of microprocessors are quite similar. In each, the state of the read-write (R/$\overline{W}$) signal at the rising edge of the $\phi 2$ (or equivalent) clock establishes whether a Memory Read or Memory Write is in progress. The memory address being exercised is signaled by decoding of the address bits to (usually) an active Low signal.

This control structure is directly compatible with the AD574A (Figure 4.13). The R/$\overline{W}$ (read/write) line can be used for R/$\overline{C}$ (read/convert); the active-low decoded base address (the AD574A occupies two memory locations) is applied to $\overline{CS}$ (chip select), and $\phi 2$ is used for CE (chip enable). The least-significant address line (even-odd) is tied to the AD574A's A0 input.

In this interfacing scheme, the processor initiates a conversion by performing a dummy Write to the converter, bringing read/convert low (the contents of the data bus during the Write are ignored). If A0 is low, a full 12-bit conversion will be started; if A0 is high, a short 8-bit conversion is started. After sufficient time has elapsed for the conversion to be complete, the processor can read the data (bringing read/convert high) in the two memory locations

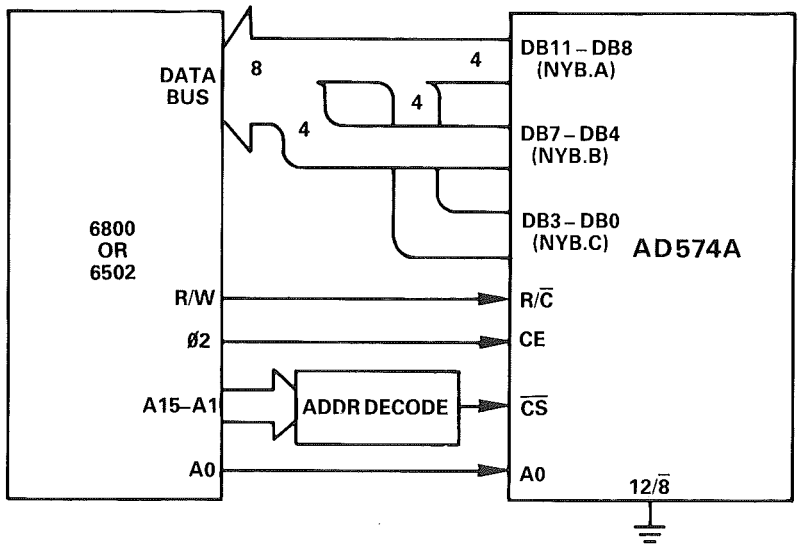


Figure 4.13. Interfacing a 12-bit ADC to 6800/6502 microprocessor.

occupied by the AD574A. The even location (A0 low) contains the 8 MSBs, and the odd location (A0 high), if used, contains the four LSBs and the four trailing zeros.

The AD574A may be used directly with 6800 series processors running at clock speeds up to 1.5MHz.

4.2.3 INTERFACING A DATA-ACQUISITION MODULE TO THE 8080 MICROPROCESSOR

Figure 4.14 shows how a DAS1128 data-acquisition subsystem (Figure 2.14b) might be interfaced with an 8080, using an 8255 peripheral interface chip. A typical sequence of events, slightly simplified, is this:

1. A setup byte, addressed to this channel (address decoded) is latched (written) into the 8255. It configures the 8255 as a set of two input ports (8 and 4 bits), which will receive the data from the converter, and one 4-bit output port, which will address the appropriate MUX channel. (The DAS1128 will have been configured for random addressing.)

2. A MUX-address byte, addressed to this channel (address decoded), appears on the data bus and is latched (written) into the MUX-address input of the DAS1128, causing the multiplexer to switch to the appropriate channel.

3. A conversion command, addressed to this channel (address decoded), is written into the DAS1128's Strobe input and initiates a conversion cycle, starting with sample-hold.

4. At some later time, when the conversion can be expected to have been completed, successive READ pulses, addressed to this channel (address decoded), cause data in the 8-bit and 4-bit input bytes of the 8255 to be transfer-

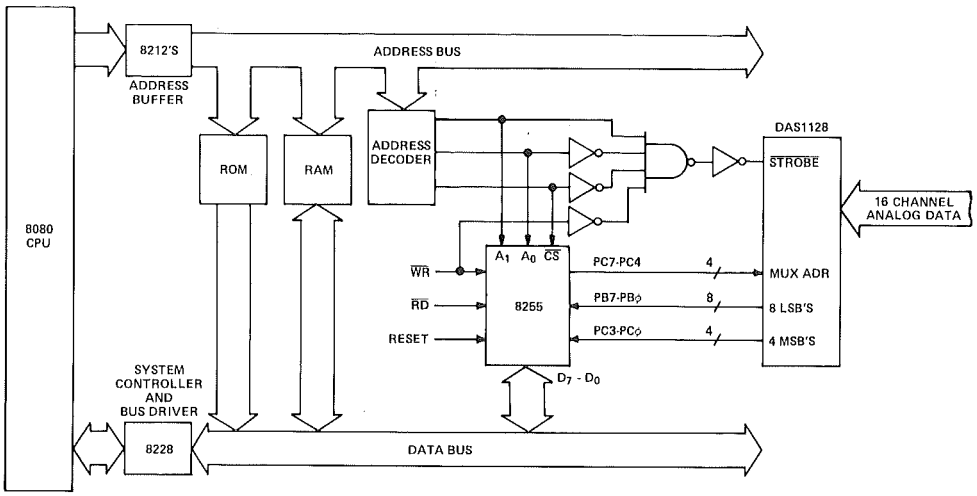


Figure 4.14. Interfacing a 12-bit DAS to an 8080 microprocessor.

red to the microprocessor. It is also possible (but not shown for the sake of simplicity) for the DAS1128's Status (Busy) line to have triggered an *interrupt* cycle when the converter's output became valid, in order to get fast handling without tying up the processor during the conversion. Interfacing could have been either I/O or memory-managed, trading off 8-bit (I/O) vs. 16-bit (memory) addressing for the sake of simpler software (2-byte instructions and faster machine handling).

It is easy to see that essentially the same technique could be used to interface any other simple 12-bit parallel-output (non-3-state) converter, such as the AD572, without the MUX, and with or without a sample/hold. If a separate sample-hold (e.g., the AD582) were used, an appropriate time delay must be used between initiating the Hold command and the start of conversion.

4.2.4 INTERFACING AN IC DAS TO THE BUS

Microprocessors. Since the AD364 data-acquisition system consists of an AD574 a/d converter plus an analog front end, the conversion control interface is the same as for an AD574. However, because there are from 8 to 16 channels of analog input, representing from 16 to 32 memory locations (for 12-bit conversion), which must be individually addressed, the decoding structure is different.

The lowest bit, A0, is still directly connected to the converter's A0 line. The next four bits, A1 through A4, are connected to Channel-Select inputs A0 through AE of the analog input section; they are internally decoded. Thus, only bits A15 through A5 on the μP's address bus need be decoded. The Channel-Select Latch is operated by the Write line; when the Write line goes low, the channel whose address is on A0 through AE of the analog input section is latched.

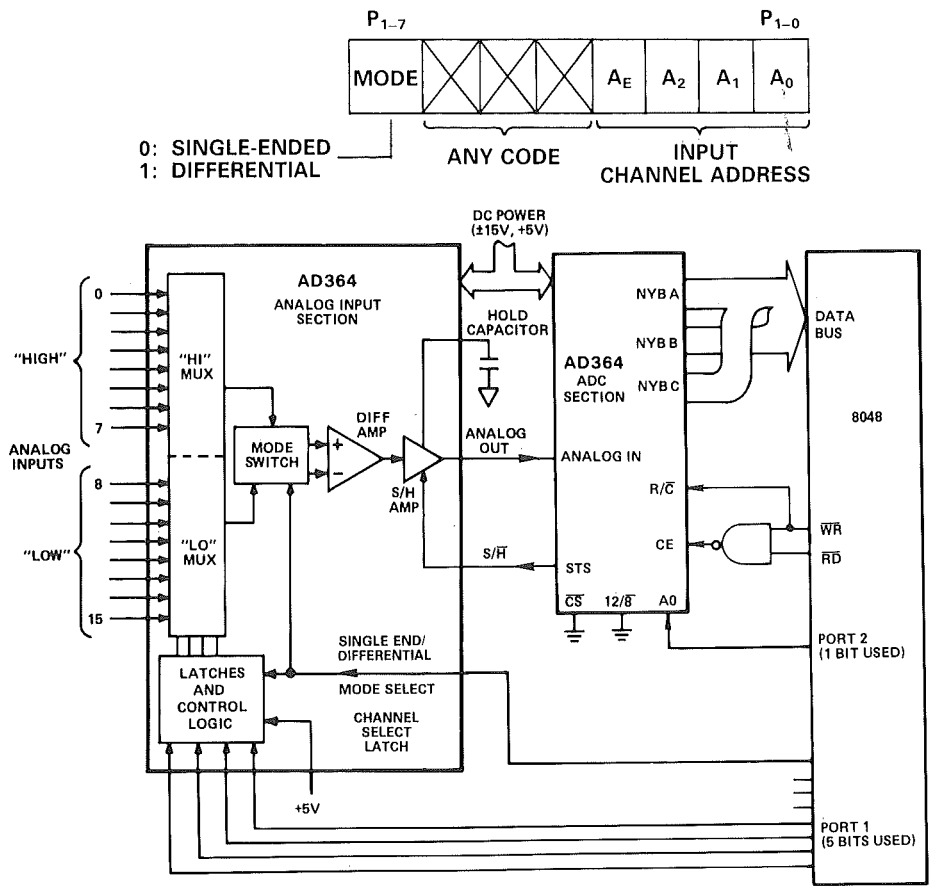


Figure 4.15. Interfacing an IC 12-bit DAS to an 8048 single-chip microcomputer.

Single-Chip Microcomputers. Single-chip microcomputers now available, such as the 8048, 6801, and 3870, include fully decoded I/O ports on the chip, as well as the central processing unit, RAM, and ROM. The fully decoded I/O ports sidestep the need for address decoding for I/O devices in many systems. For example, the 8048 contains 64 bytes of RAM, 1K bytes of ROM, and 2 programmable 8-bit I/O ports, which can be used either as inputs or outputs. A third 8-bit port, designated BUS, is a bidirectional port, which can be used for expanded I/O or memory.

As Figure 4.15 shows, the AD364 interfaces easily to an 8048 single-chip microcomputer, providing a complete data-acquisition system with minimal package count. In this system, 5 of the 8 bits of Port 1 drive the channel-select address inputs and single-ended/differential mode. Since the outputs of Port 1 are already latched, it isn't necessary to use the latch built into the AD364. The Latch input is tied to logic 1, which causes the latch to be transparent. The setup byte at Port 1 for the conversion takes on the format shown in the inset.

4.2.5 INTERFACING AN IC DAS WITH MEMORY TO THE BUS

The AD7581 (Figure 4.4) accepts eight analog inputs and sequentially converts each into an 8-bit binary word, using the successive-approximation technique. The result of each conversion is stored at an address in an 8-bit, 8-channel dual-port (input, output) RAM. Basic timing for the device is derived either from the microprocessor clock (in 6800-type systems) or from some suitable signal (ALE—Address Latch Enable, in 8085-type systems). Startup logic is included to establish the correct sequences within 800 clock periods when power is applied; required power is −10V reference and +5V excitation.

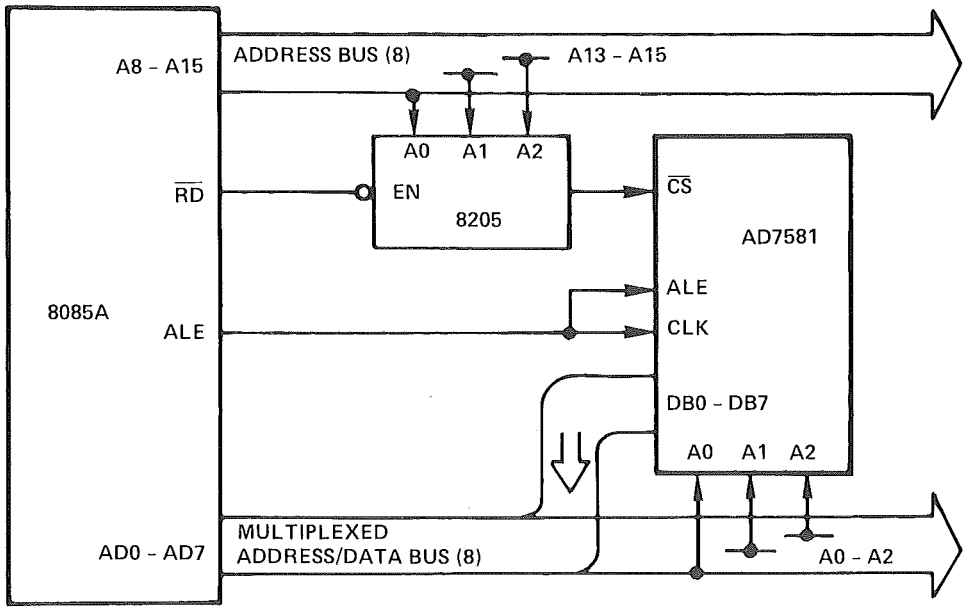
A complete conversion cycle for each channel requires 80 clock pulses; 640 pulses are needed for a complete scan through all eight channels (in decreasing order, from channel 7 to channel 0). When a channel's conversion is completed, the contents of the successive-approximation register are loaded into the proper channel location of the 8 × 8 RAM, and the Status line ($\overline{STAT}$) produces a negative-going pulse. When conversion of Channel 1 is complete, the status pulse lasts for 72 clock-periods (during the conversion of channel 0); for all other channels, the status pulse is only 8 clock periods in duration. Thus, an external pulse-width detector can be used to identify when channel 1 has been converted (and that channel 0 is being converted) and to derive conversion-related timing signals for microprocessor interrupts.

Each time the status line goes low, the multiplexer address is decremented; the next conversion starts eight clock periods later. Automatic interleaved DMA (Direct Memory Access) is provided by on-chip logic to ensure that memory updates take place at instants when the microprocessor is not addressing memory.

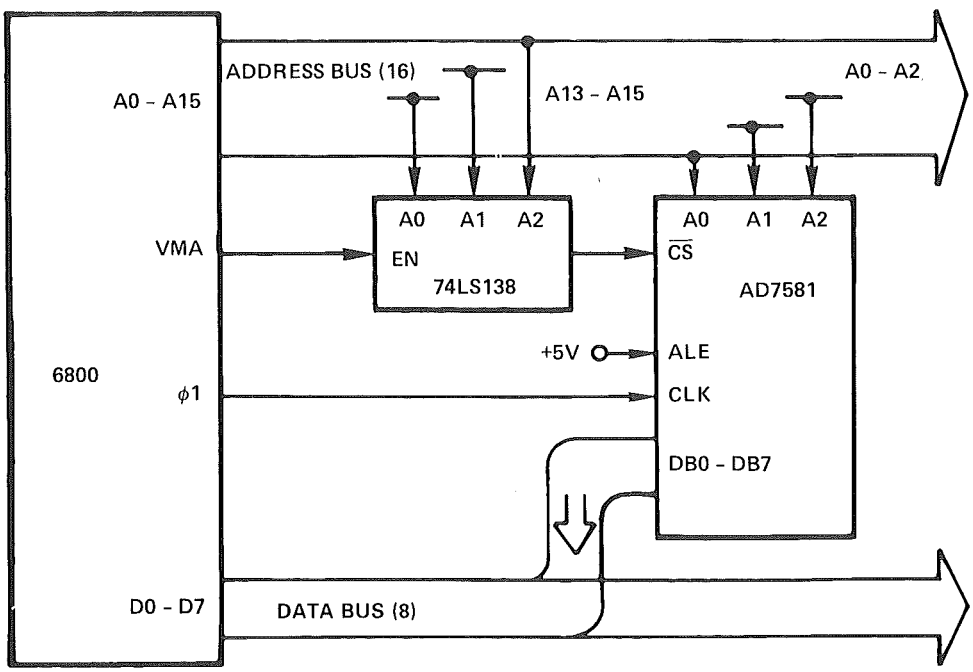
Memory locations are addressed via the three lines A0, A1, and A2. The input address latch is transparent when ALE is high and latched when ALE goes low. This address may be actively latched by ALE for systems which feature a multiplexed bus carrying both address and data information (Figure 4.16a); otherwise, for systems having separate address and data buses, the address inputs can be made transparent by tying ALE high (Figure 4.16b). $\overline{CS}$ (chip select) activates three-state buffers to place the addressed data on the DB0 to DB7 data-output pins.

4.2.6 INTERFACING ANALOG I/O BOARDS TO μC BUSES

The RTI-1260, described earlier, is one of a series of analog I/O boards, designed for complete plug-compatibility with the various microcomputer bus structures, utilizing memory-mapped architecture. In some cases, Port I/O is also available. Such cards offer the best results in the tradeoffs between hardware/software and cost/performance, with 12-bit data-acquisition throughputs of up to 30kHz.



a. 8085 interface.



b. 6800 interface.

Figure 4.16. Interfacing a memory DAS to popular microprocessors.

Memory mapping treats an analog card as a block of memory locations. The cards, which decode the addresses placed on the address bus by the microcomputer, occupy a set of (e.g., 16) consecutive bytes within an unused 1K block of addresses in microcomputer memory. The choice of configuration is user-programmed by the use of on-board jumpers, which permit a choice among 256 locations in a 64K memory space.

When the RTI cards are addressed as memory, simple memory Read or Write instructions (STA, LDA) can be used. Memory mapping also allows programmers to access the RTI boards using any of the memory-reference instructions in the repertoire (SHLD; LHLD; MOV M, r; MOV r, M). Figure 4.17 shows an example of a data-acquisition subroutine for 12-bit a/d conversion.

```
THIS EXAMPLE USES 8085 ASSEMBLY LANGUAGE TO ADDRESS
CHANNEL 1, DO AN A/D CONVERSION AND STORE 12 BITS OF
A/D DATA IN REGISTER PAIR B AND C. BASE ADDRESS HAS
BEEN SET AT FFFB.

        LXI     H,FFFB
        MVI     M,Ø1      SELECT MUX ADDRESS
        LXI     H,FFFD
LOOP    MOV     A,M
        RLC
        JC      LOOP      TEST BUSY BIT
        MOV     B,M       READ ADC DATA HI
        DEC     H
        MOV     C,M       READ ADC DATA LO
```

Figure 4.17. Example of 12-bit a/d conversion program.

RTI-1260 Analog Input Card Memory Map. The RTI-1260 uses 3 bytes, which are assigned as shown in Figure 4.18.

MUX ADDR/CONV: Any one of 32 single-ended or 16 differential input channels can be selected at random by writing the channel code into bits D0 to D4 of this byte (00000 to 11111 single-ended, 00000 to 01111 differential). Writing to this byte also initiates an a/d conversion.

A/D DATA LO: The 8 lowest-order bits of the a/d converter's output word are available at this address.

A/D DATA HI: The Busy bit (D7) is used to indicate when an ADC conversion is complete, resulting in valid data. Logic 1 indicates busy, Logic 0 indicates conversion complete. The four highest-order bits of the right-justified ADC output are also available at this address (D3, D2, D1, D0). For twos complement bipolar coding of the ADC output, the inverted MSB is read at D6, D5, D4, and D3.

BYTE ADDR.	DATA FORMAT								FUNCTION	OPER.
	D7	D6	D5	D4	D3	D2	D1	D0		
XFYB	ϕ	ϕ	ϕ	M_4	M_3	M_2	M_1	M_0	MUX ADDR/ CONV	WRITE
XFYC	B_7	B_6	B_5	B_4	B_3	B_2	B_1	LSB	A/D DATA LO	READ
XFYD	BUSY	0 / $\overline{\text{MSB}}$	0 / $\overline{\text{MSB}}$	0 / $\overline{\text{MSB}}$	MSB / $\overline{\text{MSB}}$	B_{10}	B_9	B_8	A/D DATA HI	READ

NOTES: 1. X AND Y ARE USER SELECTABLE.
2. BITS SHOWN AS ⊟ HAVE THE UPPER VALUE FOR UNIPOLAR CODES AND LOWER VALUE FOR 2's COMPLEMENT.
3. THE SYMBOL ϕ MEANS THE BIT IS IGNORED.
4. BUSY BIT EQUALS "1" DURING CONVERSIONS AND "0" WHEN DONE.

Figure 4.18. Memory map for microcomputer analog-input subsystem.

I/O Port Addressing. In the I/O port mode of operation, the RTI-1260 card occupies three consecutive ports in either an 8-bit or 16-bit port image. The port address is determined by on-board jumpers which can be configured to begin at any 16 port boundary. Since the RTI card is treated as a group of I/O ports, simple input and output instructions (INP, OUT) can be used. Port addressing eliminates the need to allocate memory when interfacing to the STD Bus.

RTI-1262 Analog Output Card Memory Map. The RTI-1262 uses 8 of the 16 contiguous bytes, assigned as shown in Figure 4.19. Since the byte addresses for the RTI-1260 and RTI-1262 are different, an input-output card pair require only a single block of memory.

D/A DATA: Two bytes are assigned to each of the four analog output channels. The digital data is written for each DAC in right-justified format, with the 8 least-significant bits in the lower address byte and the 4 most-significant bits in the upper byte. The DAC is double-buffered for one-step DAC updating; data for both bytes is loaded into the DAC only when the high byte is written to.

I/O Port Addressing. In the I/O port mode of operation, the RTI-1262 card occupies eight consecutive ports in either an 8-bit or 16-bit port image. The port address is determined by on-board jumpers which can be configured to begin at any 16 port boundary. Since the RTI card is treated as a group of I/O ports, simple input and output instructions (INP, OUT) can be used. Port addressing eliminates the need to allocate memory when interfacing to the STD Bus.

BYTE ADDR.	DATA FORMAT								FUNCTION	OPER.
	D7	D6	D5	D4	D3	D2	D1	D0		
XFY0	B_7	B_6	B_5	B_4	B_3	B_2	B_1	LSB	D/A DATA 0 LO	WRITE
XFY1	ϕ	ϕ	ϕ	ϕ	MSB	B_{10}	B_9	B_8	D/A DATA 0 HI	WRITE
XFY2	B_7	B_6	B_5	B_4	B_3	B_2	B_1	LSB	D/A DATA 1 LO	WRITE
XFY3	ϕ	ϕ	ϕ	ϕ	MSB	B_{10}	B_9	B_8	D/A DATA 1 HI	WRITE
XFY4	B_7	B_6	B_5	B_4	B_3	B_2	B_1	LSB	D/A DATA 2 LO	WRITE
XFY5	ϕ	ϕ	ϕ	ϕ	MSB	B_{10}	B_9	B_8	D/A DATA 2 HI	WRITE
XFY6	B_7	B_6	B_5	B_4	B_3	B_2	B_1	LSB	D/A DATA 3 LO	WRITE
XFY7	ϕ	ϕ	ϕ	ϕ	MSB	B_{10}	B_9	B_8	D/A DATA 3 HI	WRITE

NOTES: 1. X AND Y ARE USER SELECTABLE.
2. THE SYMBOL ϕ MEANS THE BIT IS IGNORED.

Figure 4.19. Memory map for microcomputer analog-output subsystem.

4.3 SERIAL INTERFACING

We've shown some of the ways analog information can be handled in its translation to digital and in the interfacing of the resulting digital information with a processor bus. This approach makes the most sense if the source of analog information is electrically and physically near the (host) processor. If, on the other hand, the data must be carried through an electrically noisy environment, if data and control signals must be transmitted over distances greater than a few meters, or if the data must be interfaced with terminals, communication links, or computer ports in a standard format, immediate conversion to some form of digital or pulse transmission employing a standard format and a minimal number of wires is strongly desirable.

Perhaps the simplest and most obvious approach is the use of a voltage-to-frequency converter (VFC), a device that produces a (usually asynchronous) output train of pulses or square waves at a frequency proportional to the input voltage or current (see Chapter 15). V/F converters offer high resolution at low cost, in common with other integrating methods. A v/f converter can continuously track the input signal without the need for clock pulses, convert-command signals, or any form of external logic. The direct count of its output

pulses, over a measured time period, can produce a binary or BCD digital number, which represents the average value of the input during the counting period (Figure 4.20).

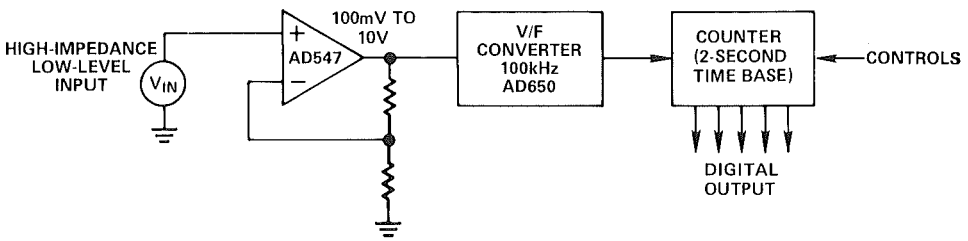


Figure 4.20. V/F converter, used as a nearly 18-bit binary (5 ½ BCD) a/d converter. Resolution is 1 pulse in 200,000, or 0.05% of smallest input signal (or 5ppm of full scale).

The VFC pulses require but a single wire-pair for transmission, unlike parallel converters, which—for n bits—require at least n + 1 wires, or synchronous serial converters, which require a form of clock signal. The v/f converter may share a local power source with a transducer and may be optically coupled for high common-mode isolation (Figure 4.21).

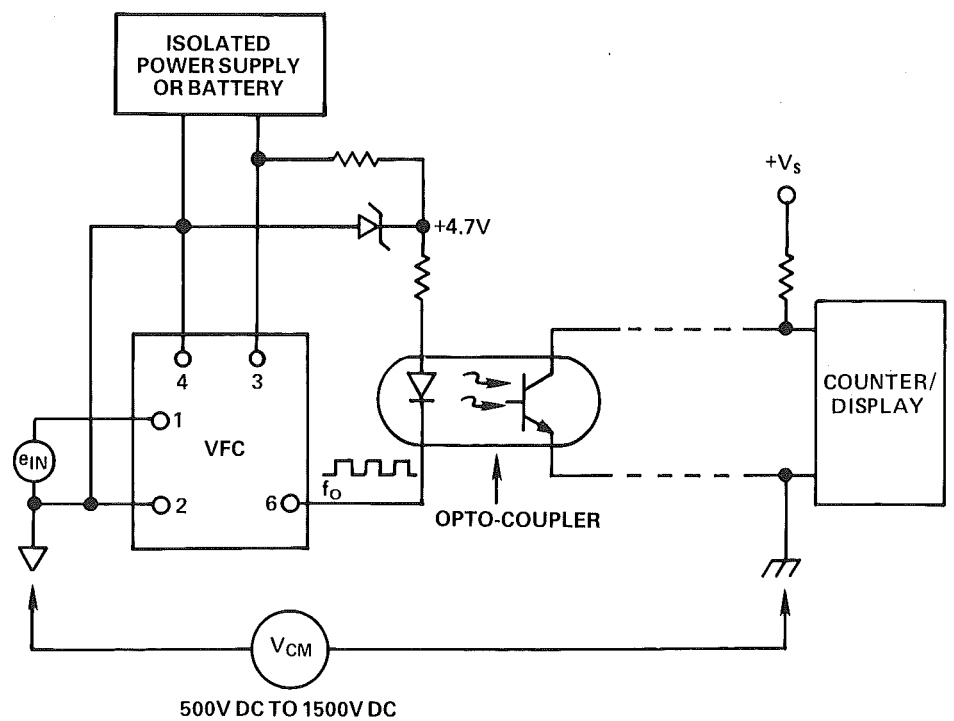
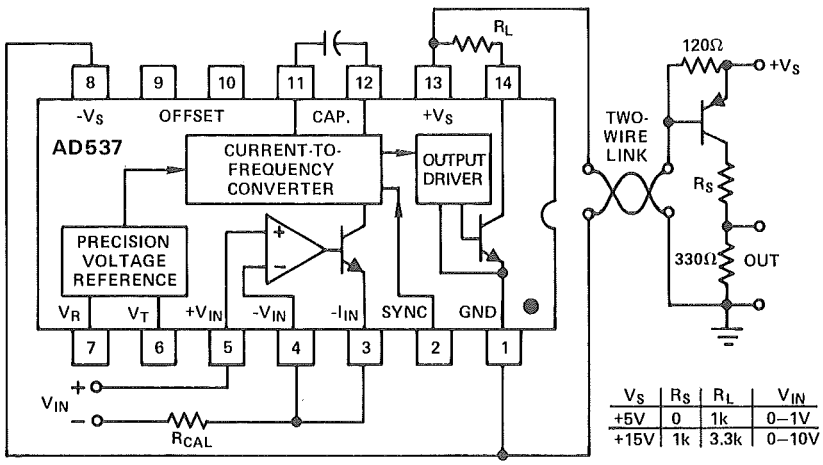


Figure 4.21. Optically isolated a/d conversion.



THE AD537 CAN BE USED FOR TRUE TWO-WIRE OPERATION, AS SHOWN HERE. THE FREQUENCY INFORMATION IS TRANSMITTED AS A CURRENT SIGNAL ON THE SUPPLY LINE TO THE DEVICE. THE SIGNAL IS CONVERTED TO A DTL/TTL OR CMOS-COMPATIBLE SIGNAL BY THE SINGLE-TRANSISTOR-TERMINATION CIRCUIT SHOWN. THE EXCELLENT SUPPLY REJECTION, HIGH OUTPUT-DRIVE CAPABILITY AND SQUARE-WAVE OUTPUT FROM THE AD537 ARE ALL ADVANTAGEOUS IN THE APPLICATION.

Figure 4.22. VFC two-wire operation.

Or a low-power-drain integrated-circuit v/f converter, such as the AD537, can use the two-wire link both to obtain its excitation voltage and to furnish an output-current pulse train, as Figure 4.22 shows. This avoids the need for local excitation. The current signal is converted to a DTL/TTL or CMOS-compatible signal by the single-transistor termination circuit shown. The excellent supply rejection, high output-drive capability, and square-wave output from the AD537 are all advantageous in this application.

The outputs from a number of AD537 VFCs may be multiplexed onto the same counter in random order by connecting their collectors together (sharing a single pullup resistor), and their emitters to the open collectors of a 1:N address decoder. Opening all gates but the one selected will cause its output pulse train to appear at the common collector terminal.

If a readily available pulse-to-fiber-optic cable driver, and the cable, are substituted for the photocoupler and transmission wires in the scheme of Figure 4.21, the VFC solves many interesting and difficult problems. These potentially include total isolation from up to millions of volts of common mode (HV transmission lines and atmospheric electricity), elimination of EMI (electromagnetic interference), privacy, light weight, small size, and—as time passes—lower cost.

VFC output pulses also lend themselves to acoustic transmission through water (or air), as well as to the modulation of RF or microwave carriers. A square-wave output, such as that of the AD537, is desirable. When using VFCs with narrow constant-width output pulses (e.g., the AD650 and ADVFC32, or the older Models 450 and 460), the output signal should be con-

verted to a 50%-duty-cycle square wave with a divide-by-2 flip flop prior to driving the ultimate transducer or modulator. This maximizes noise immunity and minimizes pulse degradation.

Although VFCs are low in cost and simple to apply in uncontrolled operation, they have shortcomings that may be serious in data-acquisition systems employing serial transmission. Principal among these is the absence of "handshaking," that is, they are not readily controlled, and their format is not very suitable for the interchange of information. Furthermore, the time required for a complete conversion cannot easily be shared for transmitting other information over the line in either direction.

Much more desirable is a means of transmitting measurements and control signals in an economical format, at will, bidirectionally over (for example) a two-wire pair to permit interfacing with data teletypewriters or other human-operated data terminals, as well as minicomputers, microcomputers, etc. This can be accomplished by the use of a standard coding and serial asynchronous word format (ASCII), and serial communication via 20-mA loops and RS-232C systems.

4.3.1 ASCII

The key to such communication is ASCII, the American (National) Standard(s Institute) Code for Information Interchange. It is the most widely used code for transmitting alphanumerics and special characters, as well as control characters of undefined appearance. It may be found in teleprinting, computing, and instrumentation. There are 128 characters in the ASCII system, transmitted in serial or parallel as a 7-bit digital word. Table 4-3 lists all ASCII characters, with their decimal and hex equivalents.

Digital Panel-Instrument Communication. An example of the use of ASCII for transmitting the results of a measurement may be found in the communication capabilities of the AD2051 "smart" digital thermocouple thermometer (Figure 4.7). It has a character-serial ASCII output and optional TTL serial or isolated full-duplex 20mA serial output.

Character Serial: In the AD2051's character-serial transmission, each character is transmitted serially to a printer or terminal on a 7-bit parallel data bus—along with a Strobe line that carries a pulse that goes low when the data for each character becomes valid. A complete transmission consists of *nine* characters, as follows:

Polarity, + or −
Four consecutive characters representing temperature (or EEEE for overrange)
Character space
Temperature scale, C or F
Carriage return signifies end of data & left-justifies column
Line feed advances printer or terminal to next line.

Character	Value		Character	Value		Character	Value	
	Decimal	Hex		Decimal	Hex		Decimal	Hex
NULL	0	00	*	42	2A	T	84	54
SOH, CTRL A	1	01	+	43	2B	U	85	54
STX, CTRL B	2	02	,	44	2C	V	86	56
ETX, CTRL C	3	03	-	45	2D	W	87	57
EOT, CTRL D	4	04	.	46	2E	X	88	58
ENQ, CTRL E	5	05	/	47	2F	Y	89	59
ACK, CTRL F	6	06	0	48	30	Z	90	5A
BELL, CTRL G	7	07	1	49	31	[	91	5B
BS	8	08	2	50	32	\	92	5C
TAB	9	09	3	51	33	]	93	5D
LF	10	0A	4	52	34	^	94	5E
VT	11	0B	5	53	35	←	95	5F
FF, CTRL L	12	0C	6	54	36	SPACE	96	60
CR	13	0D	7	55	37	a	97	61
SO, CTRL N	14	0E	8	56	38	b	98	62
SI, CTRL O	15	0F	9	57	39	c	99	63
DLE	16	10	:	58	3A	d	100	64
DC1	17	11	;	59	3B	e	101	55
DC2	18	12	<	60	3C	f	102	66
DC3	19	13	=	61	3D	g	103	67
DC4	20	14	>	62	3E	h	104	68
NAK, CTRL U	21	15	?	63	3F	i	105	69
SYN, CTRL V	22	16	@	64	40	j	106	6A
ETB, CTRL W	23	17	A	65	41	k	107	6B
CAN, CTRL X	24	18	B	66	42	l	108	6C
EM, CTRL Y	25	19	C	67	43	m	109	6D
SUB, CTRL Z	26	1A	D	68	44	n	110	6E
ESC	27	1B	E	69	45	o	111	6F
FS	28	1C	F	70	46	p	112	70
GS	29	1D	G	71	47	q	113	71
RS	30	1E	H	72	48	r	114	72
US	31	1F	I	73	49	s	115	73
sp	32	20	J	74	4A	t	116	74
!	33	21	K	75	4B	u	117	75
"	34	22	L	76	4C	v	118	76
#	35	23	M	77	4D	w	119	77
$	36	24	N	78	4E	x	120	78
%	37	25	O	79	4F	y	121	79
&	38	26	P	80	50	z	122	7A
'	39	27	Q	81	51	{	123	7B
(	40	28	R	82	52	|	124	7C
)	41	29	S	83	53	}	125	7D
						~	126	7E
						del	127	7F

Table 4.3. ASCII character chart.

In order for digital data to be transmitted on the bus, a Request input must bring the $\overline{\text{REQ}}$ line Low at a time when an output digital transmission is not occurring. If $\overline{\text{REQ}}$ remains Low, the results of each measurement will be transmitted continuously.

With the AD2051 connected to a low-cost printer, having a standard ASCII input format (Figure 4.23), an economical single-channel data logger can be assembled. A print-inhibit switch may be used to operate $\overline{\text{REQ}}$ so that only selected measurements are logged.

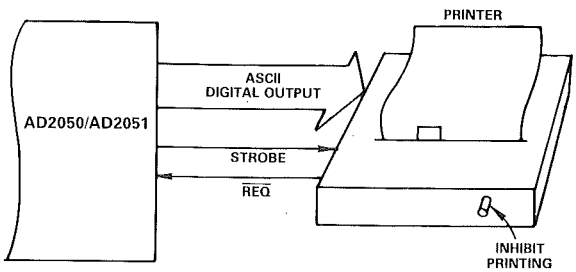


Figure 4.23. Interfacing digital thermometer to printer.

20-mA Current Loop, Bit-Serial, Full Duplex: A typical 20mA regulated half-duplex current loop is shown in Figure 4.24. A regulated 20-mA current source provides current for the loop. The current flows through receivers, transmitters, teletype machines, etc., so long as the switches are closed. When any of the devices on the line causes a switch to open, the current stops flowing, and all receivers on the line detect the level change. If a switch opens and

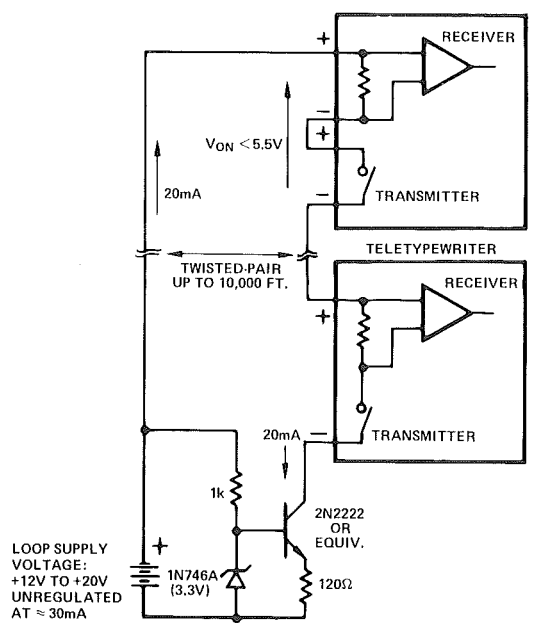


Figure 4.24. 20-mA current-loop data transmission.

closes a number of times, it will transmit a coded message to all enabled receivers; and the code, when decoded, will provide whatever information it represents and will cause whatever subsequent actions are appropriate.

A full-duplex loop consists of two current loops, one for transmitting, the other for receiving. This permits a device to receive a message while it is transmitting.

In the serial format, the bits that define an ASCII character are sent serially, as an asynchronous train of binary levels, LSB first. As noted above, the first bit of a transmission is initiated by turning off the current (0); this is the Start bit. Then, the seven ASCII data bits are transmitted (current on—1, current off—0). The data bits may be followed by an eighth (*Parity*) bit. Finally, two or more Stop bits indicate that the character transmitted has ended, and the output remains high until the next character is to be transmitted (Figure 4.25).

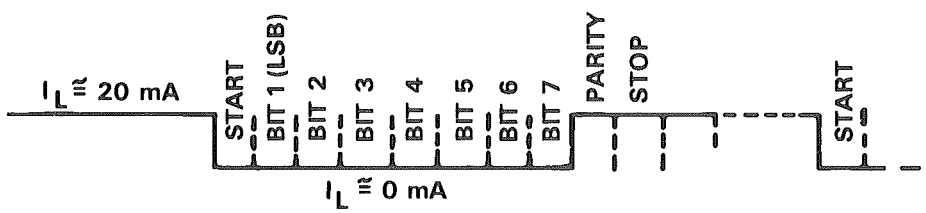


Figure 4.25. Typical ASCII code format for serial transmission.

Figure 4.26 shows how an AD2051 digital thermocouple meter communicates with a host computer or terminal using full duplex. A request for information is sent on the lower loop to the device's $\overline{\text{SERIAL INPUT}}$ terminals; the data transmission is sent back via the serial output terminals. A transmission can

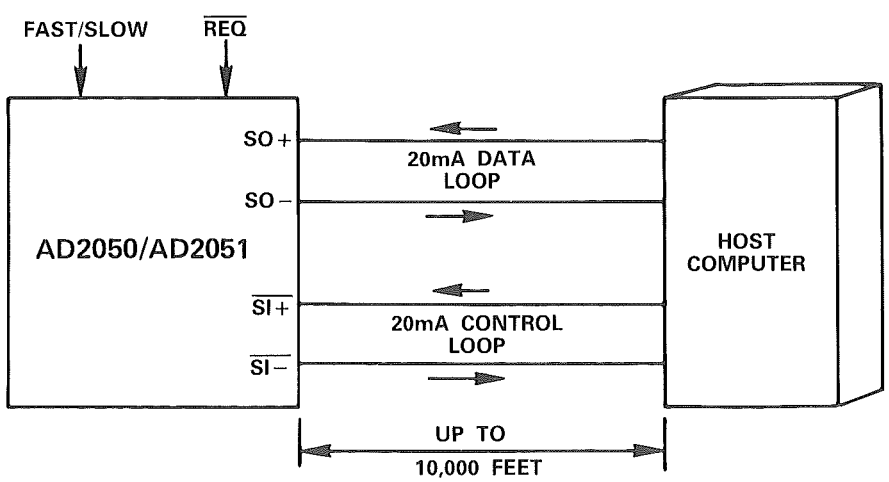


Figure 4.26. Four-wire full-duplex operation of digital thermometer.

also be initiated locally—at the meter—via the $\overline{\text{REQ}}$ line. There are two choices for the information rate, 300 baud (bits/second), or 1,200 baud. The tradeoff is information rate vs. maximum distance—10,000 feet at 300 baud, 4,000 feet at 1,200 baud.

4.3.2 RS-232C AND OTHER STANDARDS

In addition to current-loop transmission, over a minimum number of wires and fairly long distance, there is a form of voltage transmission of ASCII characters or other serial data, for distances up to 50 feet, involving a standard connector and the possibility of a variety of control and handshaking options, employing a number of additional wires.

For a peripheral device, such as a data-acquisition system, to send information back to a host computer, there are three things that must be consistent on both sides of the interface:

1. The method of encoding the information at the sending end and decoding it at the receiving end must be the same.
2. The signals used must correspond, with respect to voltage levels, timing, and sequence of signalling.
3. The connectors used must physically match.

The RS-232C Standard (C is the most recent revision), published by the Electronic Industries Association, is one of several standards* that specify some of the factors associated with these three areas; other factors are matters for agreement between companies supplying compatible equipment.

Further discussion of RS-232C is beyond the scope of this book. However, it is important for the reader to understand that, while the RS-232C standard is by far the most-common method of interconnection for computer-related electronic equipment, the term, "RS-232C-compatible" does not mean that all you have to do is plug the equipment together for everything to work. Problems can arise with the selection of parity, baud rate, and number of bits, and with the wiring of data and control lines with the up to 25 assigned wires. Syntax problems between more-complex RS-232C-compatible devices can also occur.

Compatibility with standards such as RS-232C is a desirable feature for all systems-level data-acquisition equipment that must interface with peripherals, such as terminals or printers, and host computers. MACSYM, the μMAC-5000, and the μMAC-4000, mentioned earlier, are RS-232C compatible, and the AD2051's isolated current-loop output can be translated to an RS-232C-compatible voltage format.

*RS-232C is applicable for short distances and low baud rates. For longer distances and higher baud rates, such standards as RS-422 and RS-423 are employed. For example, MACSYM 150 communicates with MACSYM 200 via an RS-422 link.

4.3.3 COMMUNICATIONS PROTOCOLS

We have seen that the AD2051 smart thermocouple thermometer communicates its output via a single 9-character word, in response to a simple digital stimulus. In the case of a multi-channel data-acquisition subsystem, which acquires analog and digital data, and can furnish analog and digital outputs to the outside world under the instructions of a terminal or a computer, the digital instructions it receives and the way it responds to them are a little more complex. In the case, of the μMAC-4000 single-board Measurement-And-Control subsystem, *communication protocols* are employed.

The μMAC-4000 measurement-and-control subsystem receives its instructions from a host computer, and communicates data upon request. Communication with the host is via a serial input/output port, which contains a full-duplex universal asynchronous receiver/transmitter (UART). The serial input-output may be jumpered for either 3-wire RS-232C communications or 4-wire full duplex.

A data-communication protocol specifies the format in which data is transferred. The μMAC-4000 employs two types of protocol ("C" and "T") to communicate with any host Computer or Terminal. The "C" protocol is designed to be used with computers and controllers, where communication efficiency, reliability, and adaptability to a wide variety of host systems are necessary. The "T" protocol, which uses simple, English-like commands, is designed for use with CRTs and teletypewriter terminals, for familiarization, debugging, system calibration, and manual control.

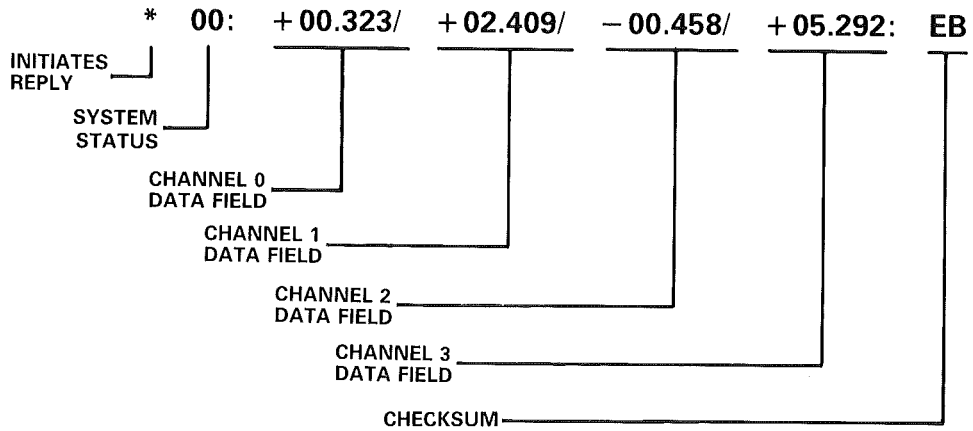
The μMAC-4000 will reply (only) in response to a command received over the serial link; this is known as command/reply (prompted) operation. Data is transmitted and received in standard ASCII format for each character, consisting of a *start* bit, 7 data bits, a parity bit, and one or two *stop* bits.

Command Sets. It is easy to operate the μMAC-4000, because the on-board microcomputer is programmed to respond to a simple command set. Through it, the host can delegate all measurement and control functions to the μMAC. The command set includes commands for transmitting analog and digital data, setting the digital output bits, activating channels, setting limits, and modifying the protocol.

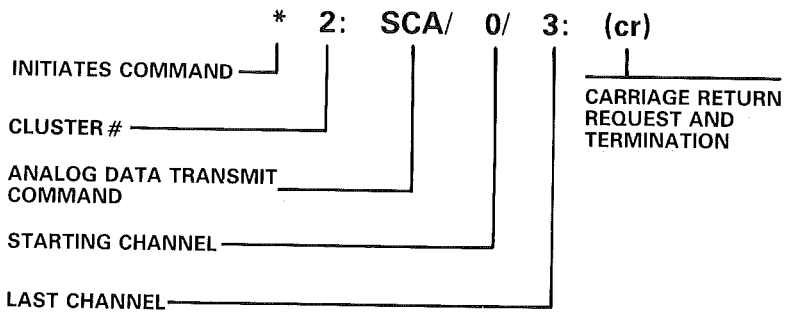
"C" Commands Figure 4.27 shows typical examples of commands (a) and responses (b).

The command instructs the μMAC-4000 to scan and transmit the latest data from channels 0 through 3 (47 channels max) of cluster (μMAC and accessories) number 2 (7 clusters max).

The carriage return of the command initiates the response. The first word is an indication of system status (errors, exceeded limits, backup power supply, etc.) Then follow the data words requested, and finally a *checksum*, the sum of the numerical (hexadecimal) values of the preceding ASCII characters



a. Typical command.



b. Typical response to the above command.

Figure 4.27. Command and response using "C" protocol.

modulo 100_H, (256). At the completion of the response, the μMAC-4000 generates a carriage return and line feed; in addition, a carriage return is also generated for each eight data fields.

"T" Commands A typical "T" command might be

CHANNEL 17 (cr)

This command requests the μMAC-4000 to transmit the latest data from Channel 17. A typical reply might be

CH 17 = +0024.7

If a thermocouple is connected to this channel, this response might indicate a temperature of +24.7°C.

The key feature of this protocol scheme is the small amount of software necessary to support an intelligent system. It permits easy implementation by hosts using high-level languages (FORTRAN, PASCAL, BASIC), and it is easily

debugged. Software drivers, available for a variety of popular computers, make it unnecessary for the user to write special programs. For example, when the appropriate software drivers are used (Model AC1820) the list in Figure 4.28 contains (1) the entire APPLESOFT BASIC subprogram to read a temperature at one location and (2) the response to the RUN command.

```
LIST 80,130

80 TN$ = "0":TR$ = "3"
90  GOSUB 61080: REM  INITIALIZE VALUES
100 TN$ = "1": REM  SELECT CHANNEL 1
110  GOSUB 61260: REM  INVOKE CHANNEL 1 READ
120  PRINT "TEMP1= " + TQ$ + " DEGREES C":
     REM  DISPLAY RESULTS
130  END
]RUN

TEMP1= +0025.4 DEGREES C
```

Figure 4.28. Display of program LIST and results of RUN, using Apple II computer and software drivers with single-board measurement-and-control subsystem.

Since the μMAC-4000 retains the updated data in memory, the response appears immediately after the RUN is executed. A scan of a number of temperatures is almost as simple and as quickly executed.

4.4 CONCLUSION

We have summarized in this chapter a variety of means of implementing converter-interface functions in terms of standard readily available products, at the various levels of system involvement that are likely to concern our readers. More information on any aspect of this discussion is no farther away than the manufacturer's nearest applications engineer.

Chapter Five

Analog Functions with Digital Components

The world of analog system designers employs numerous circuit tricks to perform operations on voltages and currents—with op amps, multiplier/dividers, filters, phase shifters, function generators, etc. The availability of converters and fairly simple digital hardware and firmware greatly enlarges this bag of tricks (without even including the contributions of computers, software, etc., which are considered in great panoply elsewhere in the book).

The term "analog" is commonly understood in two contexts, one valid and one less-than-convincing: "analog" in the sense of dealing with measurable real-world quantities rather than abstract *digital* numbers; and "analog" in the sense of smooth, or at least *continuous* (derivatives existing nearly everywhere), vs. *discrete* (sampled or quantized), signals or relationships between inputs and outputs. We contend that, when used in the latter sense, the term *discrete* also connotes analog (measurable) quantities, but in a bridging state between analog and digital.

There have been a few excellent books on the applications of operational amplifiers, fewer on the applications of op amps and analog function modules, and virtually none on the use of digital and interface components (converters, counters, shift registers, etc.) in the service of analog relationships.

There are many excellent auguries favoring an intimate, long, and happy marriage between the two tribes. Analog devices are cheap, plentiful, and capable of a great deal of functional versatility; an increasing number of digital devices (including CPUs) are also cheap, plentiful, and capable of a great deal of functional versatility. The reasons there has been little apparent intercourse between them are twofold. Interface devices, such as A/D and D/A converters have heretofore been too expensive to be wasted as components (old-timers

will remember the days of $227 op amps and $50 transistors). But more important—practitioners who volubly embrace the tricks of both trades have remained either extremely rare or well-hidden.

This chapter is in no sense intended as an encyclopedia (in either breadth or depth) of such connubial (i.e., "hybrid") circuits; that volume is yet to be written. Rather, the few representative items included here are intended to be suggestive of what is possible, and to stimulate the reader to bring creative faculties to bear on new ways of looking at problems that may have been conceived of as being strictly "analog" or "digital." For those already laboring in the vineyard, there will be no revelations, but perhaps there is something a little new or different to make a scan worthwhile. The circuits are presented in the form of independent modular panels that stand alone ("bite-size morsels," to aid digestion). The selected examples are:

SOURCES

- Digitally-Controlled Voltage Source
- Manual Digital Inputs
  - Thumbwheel BCD switch
  - Toggle-switch register
- Digitally-Controlled Current Sources
  - "Current-output" DAC
  - Current gain: floating load
  - Current gain: buffered load
  - Current to grounded load
  - 4-to-20-mA Current Generator

SCALE FACTORS AND MODULATIONS

- Digitally-Controlled Direct Gains
- Digitally-Controlled Inverse Gains
- Logarithmic Scale Factors
- High-Precision Analog Multiplication
- . . . or Division

FUNCTIONAL RELATIONSHIPS

- Analog Functions with Memory Devices
- Sinusoidal Input-Output Relationships

TRIGONOMETRIC APPLICATIONS

- Digital Phase Shifter
- Digital/Resolver Converter (Resolver Simulator)
- Coordinate Conversion

5.1 SOURCES

DIGITALLY CONTROLLED VOLTAGE SOURCE
(or Precision Power Supply)

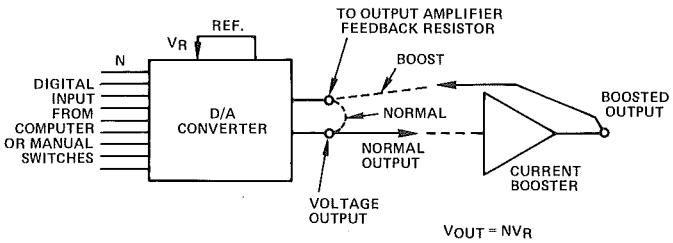
A well-calibrated internally referenced d/a converter is probably the simplest available source of arbitrary precision voltages. Turn on the power, set the digital input, and expect (and receive) the voltage you asked for. With a 10-bit converter, resolution is 0.1%; with a 12-bit converter, 0.024%; and with a 16-bit converter, 0.0015% (15 ppm).

Let it be driven by a computer, and you have a ready supply of voltage for fast or slow automatic testing, Set it manually (with a "toggle-switch or DIP-switch register," or with BCD thumbwheel switches), and it's a convenient "volt-box," or a handy reference source. Or set it permanently by hard-wiring its logic inputs. No resistors or pots necessary!

If its output op amp doesn't have adequate output current, follow it with an inside-the-loop current booster. Feedback to the built- in amplifier-feedback-resistor will make the output virtually independent of the booster's dc characteristics. It can be followed with an op amp having higher-voltage output and precisely set fixed gain, if high voltage is needed. Doing this outside the DAC's loop protects the converter's circuitry (including the low-voltage digital components) from accidental exposure to fault voltages.

Because the setting is done digitally, in either serial or parallel, the voltage can be set from a distant location, or in the presence of a fair amount of electri-

cal noise, relying on the inherently high noise immunity of digital signals (at the cost of additional wire for parallel circuits). If noise pickup is not a major factor, it is interesting to note that in some cases the switches can be closed "passively," i.e., to the power-supply return for "0", left open for "1".* The serial-input AD7543 may permit remote voltage (or gain) settings, with minimal wiring, when appropriately pulsed.



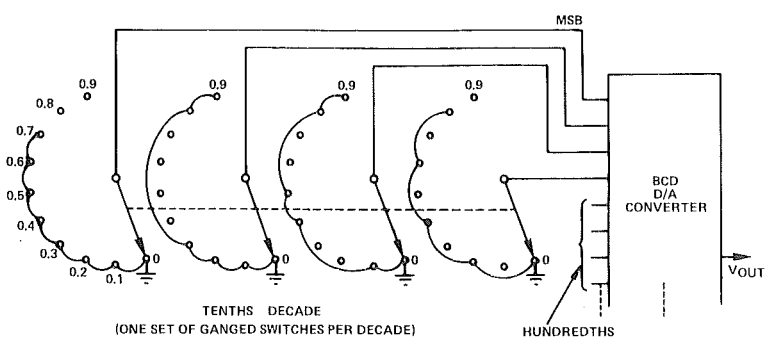
MANUAL DIGITAL INPUTS

All that is needed to obtain a given output voltage from a D/A converter is to close the appropriate switches. Human beings usually prefer base-10 numbers or BCD coding, despite the fact that it throws away inherent binary resolution at the rate of 2-bits-out-of-12 (12 BCD = 1/1000, 10 BIN = 1/1024).

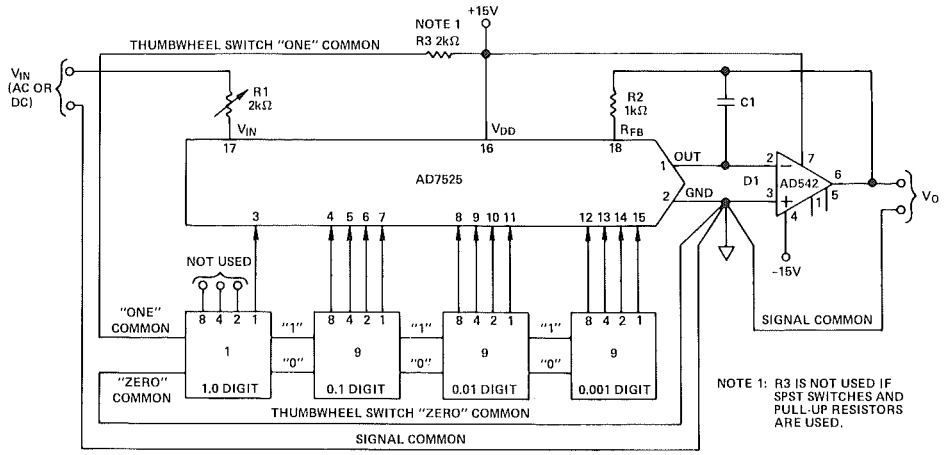
Thumbwheel-Switch Encoder

A thumbwheel-switch encoder is the simplest way for the operator, especially one who is mathematically unsophisticated, since the base-10 number can be set directly, and all the appropriate switches are automatically closed. A D/A converter with BCD coding should be used. The switch points that are "0" (positive true) are connected to ground; those that are "1" are either left open* or connected to $+V_S$ (but be sure to use a break-before-make switch).

The first figure shows the principle for one decade of thumbwheel switchery ("1" open). If the converter has *complementary BCD* coding, the complementary switch connections should be used. A complete circuit arrangement employing a BCD d/a converter is also shown.

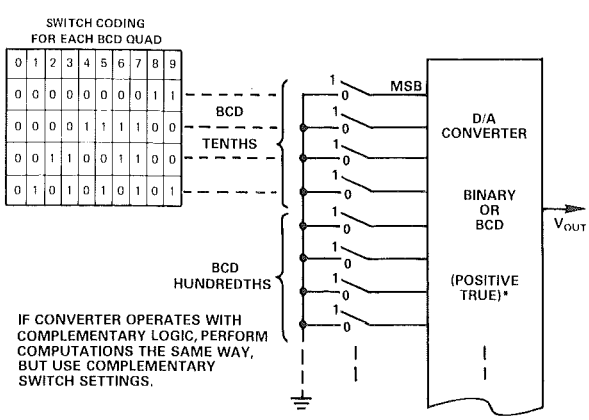


*"Pull-up" resistors, connected between each digital input and the positive supply, ensure that when the switch is open, the input to the DAC will indeed be "1".



Toggle-Switch Register

The toggle-switch register is physically more elementary, and it may be used with either binary or BCD-coded DACs. It does require some calculations, though, especially for binary settings. As an aid to calculation, two tables are given, one for BCD (the same table is used for each digit), and one for binary equivalents of representative decimal fractions of full scale. Interpolation is performed by adding or subtracting an appropriate set of terms (binary rules) to form the desired sum (for example, $0.52 = 0.5 + 0.02 = 0.1000\ 0101\ 0001\ 1110_2$.) Note that multiplication or division by 2 simply moves a number one place to the left or right: by 4, two places left or right, etc.



For unipolar binary coding, the digits to the right of the "decimal" point form the code, MSB leftmost. For bipolar 2s complement, divide the magnitude by two for the positive number, then complement all digits and add 1 LSB for the negative number. For offset binary, complement the 2s-complement MSB. (See Chapter 7 for a more-complete discussion of coding and conversion relationships in bipolar DACs.)

BINARY EQUIVALENTS OF DECIMAL FRACTIONS

	MSB ↓				
0.8	0.1100	1100	1100	1101	0
0.5	0.1000	0000	0000	0000	0
0.4	0.0110	0110	0110	0110	1
0.25	0.0100	0000	0000	0000	0
0.2	0.0011	0011	0011	0011	0
0.125	0.0010	0000	0000	0000	0
0.1	0.0001	1001	1001	1001	1
0.08	0.0001	0100	0111	1010	1
0.0625	0.0001	0000	0000	0000	0
0.04	0.0000	1010	0011	1101	0
0.02	0.0000	0101	0001	1110	1
0.01	0.0000	0010	1000	1111	0
0.008	0.0000	0010	0000	1100	1
0.004	0.0000	0001	0000	0110	0
0.002	0.0000	0000	1000	0011	0
0.001	0.0000	0000	0100	0001	1
0.0008	0.0000	0000	0011	0100	1
0.0004	0.0000	0000	0001	1010	0
0.0002	0.0000	0000	0000	1101	0
0.0001	0.0000	0000	0000	0110	1

Converting Base 10 Number to Binary Switch Setting — Two Examples (12-Bit Conversion):

1. +0.9FS (=0.5FS+0.4FS)

0.5	0.1000	0000	0000
+0.4	+0.0110	0110	0110
0.9	0.1110	0110	0110

Code: 1110 0110 0110, Straight Binary

2. −0.6FS, 2s Complement (Note: 0.6 = 0.5 + 0.1)

0.5	0.1000	0000	0000	
0.1	+0.0001	1001	1001	
0.6	0.1001	1001	1001	
Code:	1001	1001	1001,	Straight Binary
x 1/2	0100	1100	1100	Scale Expansion
Compl.	1011	0011	0011	Ones Complement
+1 LSB	1011	0011	0100,	Twos Complement

DIGITALLY CONTROLLED CURRENT SOURCES

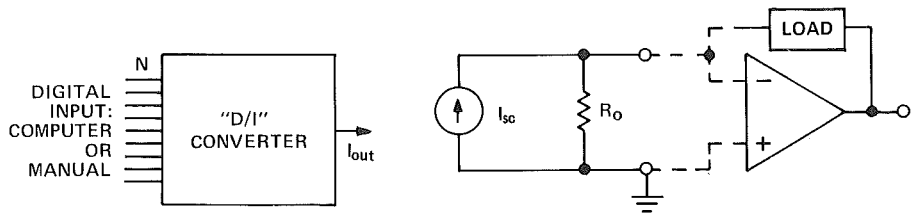
Many analog current sources have been developed with the variations that provide such diverse advantages as low cost, simplicity, ability to transmit analog information over distance with reasonable noise immunity, ability to ground the load, etc. In conventional all-analog circuits, the original controlling input, if constant, is derived typically from a precision potentiometer, zener diode, or other reference—and if variable, from a sensor or signal conditioner. However, availability of versatile D/A converters now permits convenient digital control of current values, making, for example, programmable current supplies and digitally controlled current transmitters an inexpensive reality. As with voltage sources, the adjustments may be performed by either a computer or a human operator. They may be purchased as complete entities (see Figure 3.14 for a sophisticated example) or wired by the circuit designer. These are a representative few among the many ways of accomplishing current drive.

"Current-Output" DAC

An ordinary d/a converter specified as a current-output DAC would appear to be the simplest form of digital-to-current output source. However, most such devices are unsatisfactory as current sources because they generally have appreciable internal admittance "looking back," and this admittance (and the load) must be included in computations of the share of current reaching the load. In addition, there may be stringent limitations on voltage swing (*compliance* voltage). For this reason, current-output DACs are almost always used with the load in the feedback path of an op amp. The DAC drives the inverting input terminal, which is normally at zero potential, thus imposing negligible loading error. However, devices such as the AD561 (see Figure 9.14), with active-collector current outputs, may be treated as true current sources over the rated compliance-voltage range.

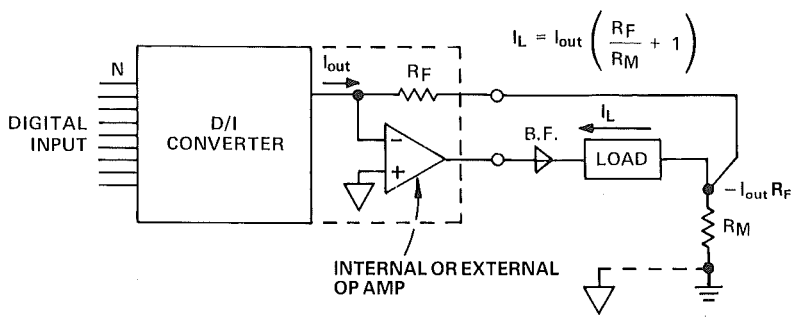
The output resistance of DACs made up of quad current sources is often introduced by the resistive dividers used for attenuation of less-significant-bit currents (see Figure 9.11 and accompanying text). For applications in which

a restricted number of discrete values of current (say 16, at equal intervals) are required, one can construct a highly precise fast current-output converter with high internal resistance, using a quad current switch.



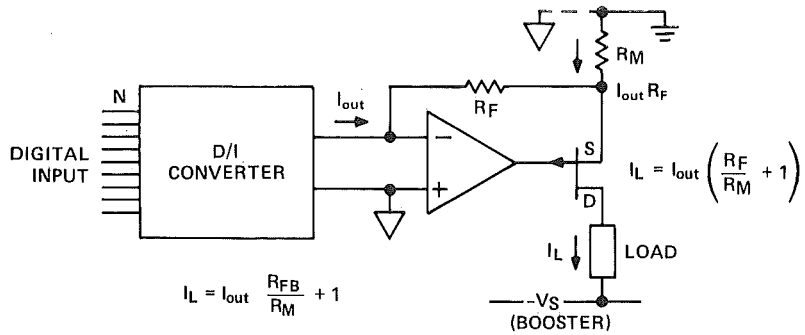
Current Gain – Floating Load

In this application, a load that has both terminals available is connected between the amplifier output terminal and the return lead of the feedback resistor. The attenuation introduced by R_M, if used, produces current gain. If the amplifier's output current is inadequate, a unity-gain current booster may be used, inside the loop (BF). For large currents, a separate booster supply should be used, with only the R_M pickoff point connected to the converter's analog ground.



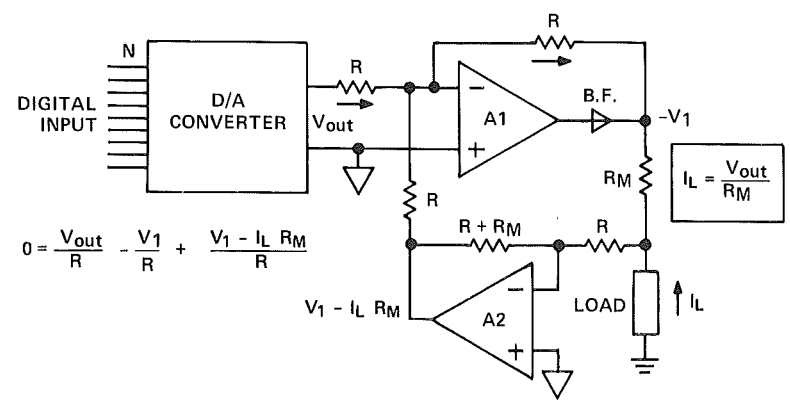
Current Gain – Buffered Load

For applications in which the amplifier's output range imposes serious restrictions on the kind of load that might be driven, a field-effect transistor (FET) with the load in its drain circuit allows a wide range of voltage swing (compliance voltage) for the load. Examples of loads that might be driven in this manner are CRT deflection coils, motor windings, chart-recorder pen drives, etc.



Current to Grounded Load

There are a number of ways of driving current to a grounded load, all of which employ both positive and negative feedback to measure and control the current. One example, using a voltage source and two operational amplifiers, is shown here. Amplifier A1 measures the difference voltage across R_M (direct from the top and inverted from the bottom via A2) and sets it equal to the DAC's V_{out}, thus forcing a current V_{out}/R_M through the load. In the general case, the resistor ratios can be adjusted for scaling, the drive could be from a current source, boosters could be used (at point "BF") etc. As with all operational-amplifier circuits having complicated (or even simple) dynamics, attention should be paid to dynamic stability: feedback capacitors may not be as helpful as capacitance shunting the load.



4- to 20-mA Current Generator

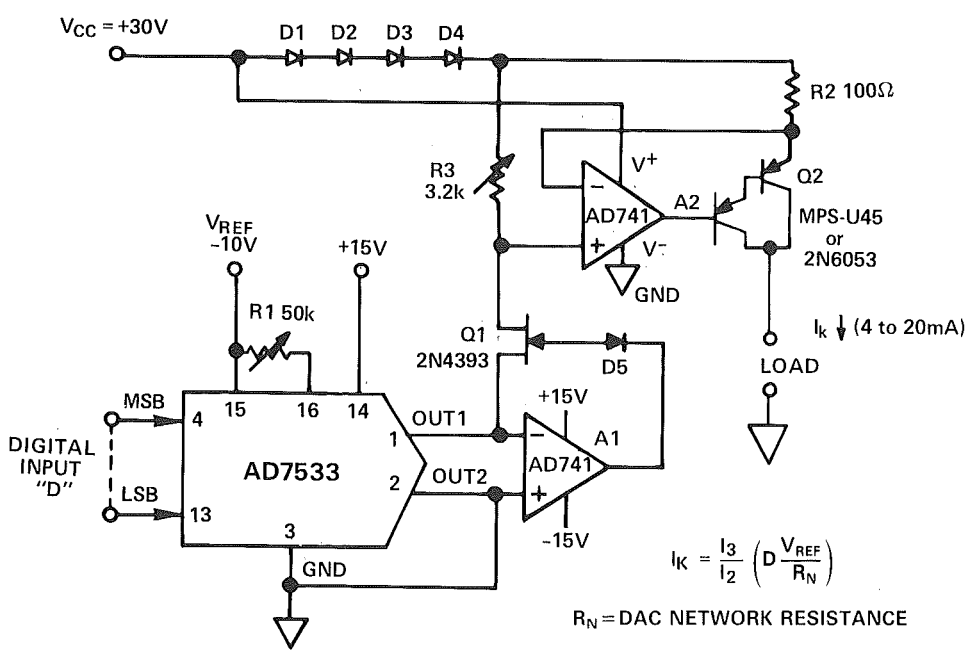
The current loop is a popular way to avoid the effects of voltage drops and minimize noise-voltage pickup due to interference when transmitting signals over moderate distances. The circuit on the next page illustrates one way to generate a digitally controlled 4-to-20mA current that is independent of the load resistance. The output current is equal to 4 mA + (D × 16 mA), where D is the fraction of full-scale output range represented by the digital input.

With a 10-bit DAC, such as the AD7533, the circuit provides an output range from 4 mA to 20 mA, with a resolution of 15 5/8 μA. The maximum compliance voltage of the load is +25 V, equivalent to a load resistance of 1,250Ω maximum.

The DAC's output current flows through Q1 and R3. Operational amplifier A2 holds the lower end of R2 at the same voltage as the lower end of R3. Since the upper ends are connected together, they have equal voltage; therefore, the current through R2 is equal to the current through R3, amplified by R_3/R_2. The current through the Darlington follower and the load is very nearly equal to the current through R2.

Low-frequency voltage variation in the 30-volt supply does not significantly affect the current through the load, since A1 adjusts Q1's output conductance to keep the lower end of R3 at whatever voltage is necessary to maintain the current through Q1 equal to the output current of the DAC.

R1 is used to adjust the ratio of full-scale to zero-scale current (at OUT 1) in the ratio of 5:1. R3 adjusts circuit offset and span to provide proper values of 0 and full-scale current, set with a given value of R_L. Diodes D1–D4 limit the common-mode voltage to A2, and diode D5 protects Q1 during power sequencing.



5.2 SCALE FACTORS

DIGITALLY CONTROLLED SCALE FACTORS

A D/A converter that accepts variable references (i.e., a *multiplying* DAC) can be thought of as a digitally-controlled potentiometer or a programmable-gain amplifier/attenuator. As such, it can be used for setting gains, either by a computer or a human operator. Computer-setting might be used, for example, in digital audio or adaptive control systems; manual setting might be employed where the device being controlled is remote (think of it as a potentiometer with a long shaft).

The multiplying D/A converter can also be thought of as a means of modulating a computer output by an analog signal. For example, if the computer is developing a square wave, the analog signal might be amplitude-modulating it (such an application might also be thought of as digitally weighted sampling).

The simplest device operates in one quadrant, with either a positive or a negative analog signal and straight binary or BCD coding. While CMOS DACs will do this easily, even many current-source DACs, such as the AD566A, which are usually unipolar, will accept a reasonably wide signal range (+1V to +10V) without excessive degradation of linearity.

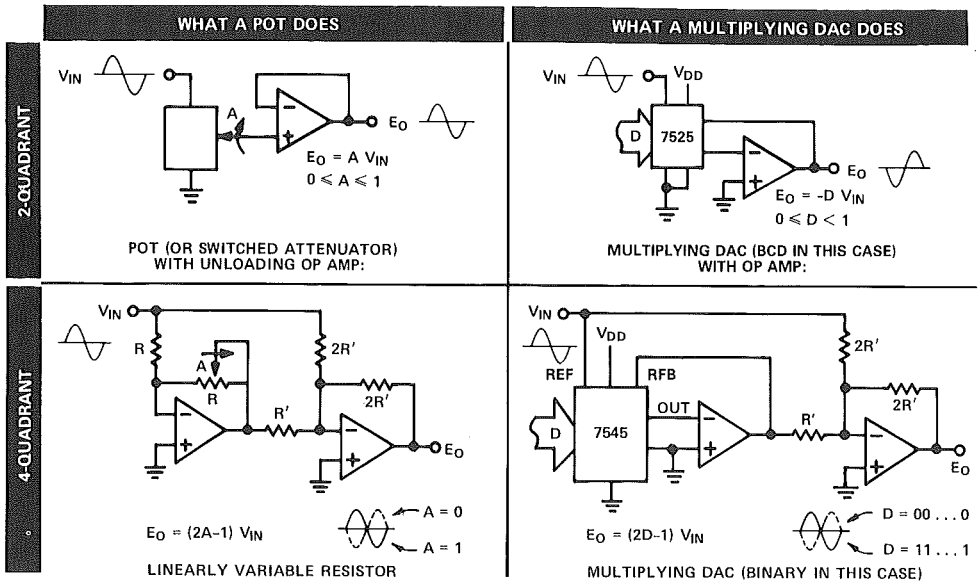
For two-quadrant operations, there are two modes: bipolar analog and bipolar digital. Bipolar analog operation simply requires a bipolar analog input and straight binary or BCD digital coding. It also requires a converter that can accept analog signals of either polarity. CMOS DACs, such as the AD7545 in the current-switching mode, are well-adapted to this form of operation, providing wide bandwidth, good linearity, and low feedthrough. "Feedthrough" is the proportion of analog input signal that appears at the output when the digital input is calling for zero gain.

Bipolar digital operation can involve offset-binary (or twos complement) coding, with the output offset by one-half of the full-scale span; or sign-magnitude coding (unipolar DAC), where the sign-bit switches the output polarity between the direct output and an inverted version of it.

Four-quadrant operation involves a combination of circumstances: a DAC that can respond to both bipolar analog and bipolar digital inputs in the correct polarity, with appropriate speed and feedthrough performance. Again, CMOS DACs in the current mode, with an additional inverting op amp, perform this function with good linearity and speed, and low feedthrough.

Shown here are five basic ways (among many) that digital gain control can be used to perform useful functions.[1]

[1]See also *CMOS DAC Application Guide*, by Phil Burton, Analog Devices (1984).



Direct Scale Factor

These "digital potentiometer" circuits provide simple linear digital scale adjustment, proportional to the unipolar or bipolar digital number. The comparison with analog potentiometer circuits is striking. As noted earlier, the digital number can be applied either by a computer signal or manually.

Inverse Scale Factor (Hyperbolic Gain Function)

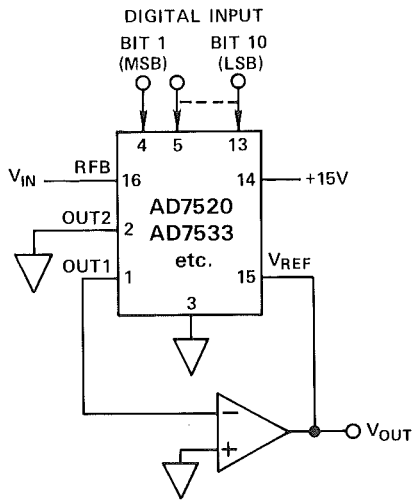
With the DAC in the feedback loop of an operational amplifier, the gain is inversely proportional to the digital number (Gain = 1/D, where D is the fractional value of the digital number). The illustration shows a CMOS DAC, in the inverter connection. When the input is connected via the DAC's internal feedback resistor, the minimum gain is nearly unity, 1/(1 – LSB). Resistance may be added in series with the input for attenuation, so that normalized unity gain can occur at a mid-scale value (for example, if the total input resistance is $16R_f$), the gain will be 1/(16D).

This circuit will amplify signals; however, it will also amplify noise and gain errors, and bandwidth will be reduced as the gain increases. Also, there are constraints required by the nature of feedback: the digital input must be unipolar and must not go to all-zeros; however the analog input may be bipolar.

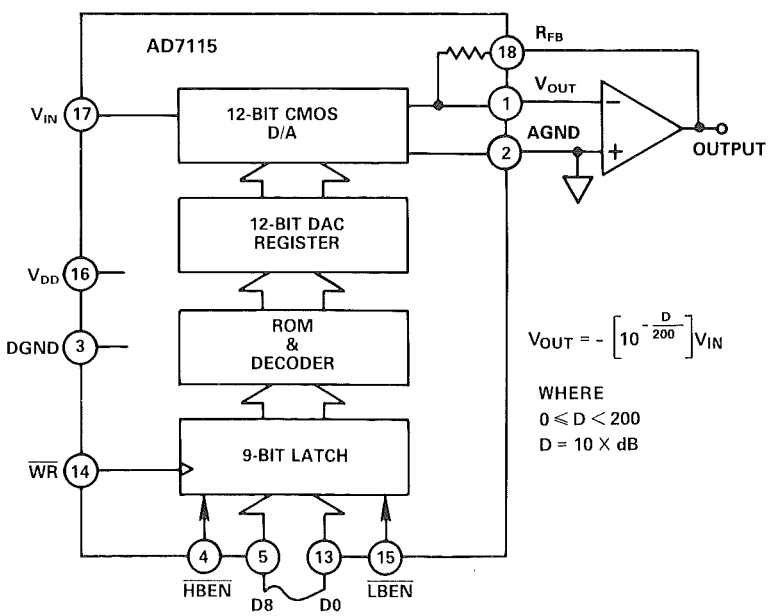
Logarithmic Scale Factors

For the many applications where logarithmic gain control is desirable (for example, dB steps of audio gain), a *logarithmic DAC* may be used. A logarithmic

D	NOMINAL $\frac{V_{OUT}}{V_{IN}}$
1111111111	$-\frac{1024}{1023}$
1000000000	-2
0000000001	- 1024
0000000000	OPEN LOOP



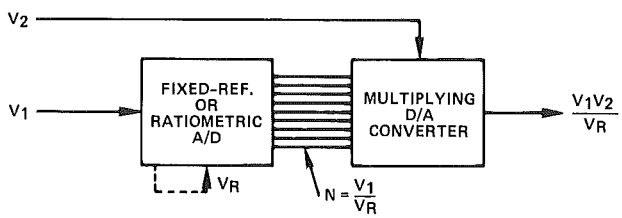
DAC (see also Chapter 16) is one for which the digital word is proportional to the logarithm of the gain, i.e., each one-bit digital change corresponds to a fixed ratio of gain change, hence a constant number of dB (20 $\log_{10}$ R). For example, the AD7115 adjusts gain in 0.1-dB steps over a 20-dB range, while the AD7118 adjusts gain in 1.5-dB steps over a nominal 88.5-dB range. In the forward connection, attenuation is adjusted, and in the feedback connection, gain. The illustration shows the analog connections of a 2 ½-digit (0 to 199) BCD DAC that can be used for microprocessor controlled attenuation of up to 20dB in 0.1-dB steps.



High-Precision Multiplication

Since a 12-bit multiplying DAC develops accuracies to within considerably better than 0.1%, it is possible to make an analog-to-analog multiplier having excellent accuracy by converting one of the inputs to digital form and using it to control the gain of a multiplying DAC. If the ADC is ratiometric, the output is a function of three variables. V_R should always be larger than V_1, or else overrange indication will be necessary. A sample-hold may be necessary if V_1 varies rapidly.

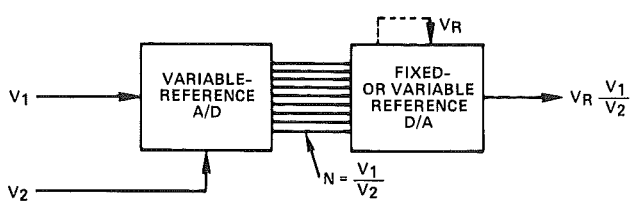
The availability of fast, high-resolution digital multipliers (e.g., 16 bits × 16 bits) may suggest the possibility of performing analog-to-analog multiplications at unheard-of speeds and accuracies. This is feasible in principle, but when the costs of sample-holds, conversion devices, and the multipliers themselves are added up, it makes more sense to consider such operations in a system context—as discussed under Digital Signal Processing—rather than as a restricted *ad hoc* circuit to perform a localized function with emphasis on simplicity and cost, which is the focus of this chapter.



...or *Division*

There are at least two approaches. In the first, the same scheme is used as for multiplication, but the DAC is used in the feedback path of the output amplifier, thus dividing the multiplier input (V_2) by the digital number representing V_1. Alternatively, since an A/D converter digitizes the ratio of the "input" to the "reference", the digital word will be the quotient and a D/A converter will convert the ratio back to a voltage. Again, if the D/A is a multiplying type, the output is a function of three variables.

For both of these applications, the A/D may be connected for either clocked or free-running operation, and either the A/D or the D/A should have a register to store the previous value and buffer the D/A during the conversion process.



5.3 FUNCTIONAL RELATIONSHIPS

The term "functional relationship" (between two voltages/currents) implies a black-box causal operation, $y = f(x)$, $f(\ )$ being any single-valued linear or nonlinear realizable function. It is distinguished from a "function generator," which implies a *time function*; in a function generator, $y = f(t)$. By applying a linearly-increasing function of time to a device having a given functional relationship, one can create a function generator.

In analog circuitry, functions are traditionally embodied in three ways:[2]

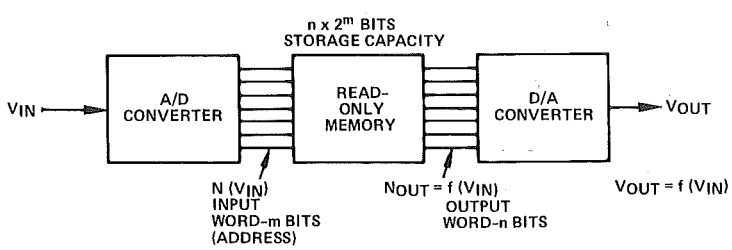
1. Using a natural function (e.g., the inherently logarithmic diode characteristic for log and antilog circuitry, the transconductance relationships of transistors for transconductance multipliers, the ability of a capacitor to store charge for integration).

2. Using diode-resistor networks to form piecewise-linear approximations to a nonlinear function.

3. Using combinations of natural functions to approximate arbitrary relationships; for example, power series using multipliers to generate the x_2, x_3, x_4, etc., terms.

Now that converters and memories are available at low cost, a fourth approach becomes feasible:

4. Using memories (e.g., ROM's singly or in groups) to store a function digitally, and converting-in and -out with clocked or free-running A/D's and D/A's, as shown in the illustration. Typical applications include trigonometric transformations, thermocouple compensators, and arbitrary functions.



Arbitrarily Programmable Functional Relationships

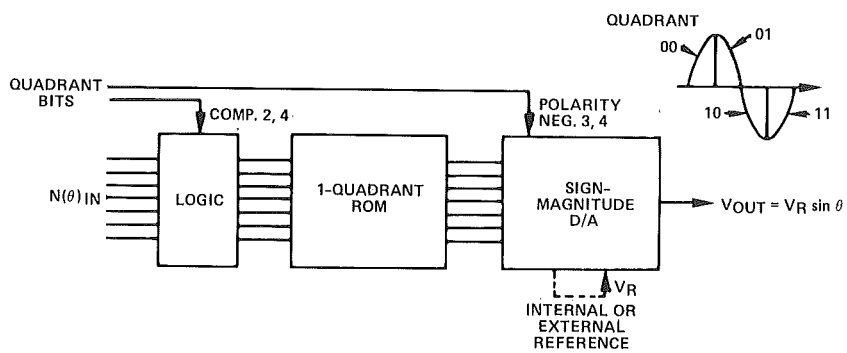
Functions can be purchased in ROMs or programmed ("burned") into the various classes of programmable read-only memories (PROMs), and used in the relatively simple manner shown to provide continuous functional relationships in real time.

[2]Nonlinear Circuits Handbook, Analog Devices, Inc. (1974), has many details of these methods.

This approach may be favorably compared with a system approach that might go something like this: the same analog-to-analog functions, in wide variety, may be performed in software with computer programs written to acquire data (V_{IN}), using an ADC at an appropriate level of integration, retrieve the stored equivalent value from memory (or compute it on the spot), and output it to the DAC (V_{OUT}). If the ability to follow a rapidly varying input continuously in real time is desired, this can be unwieldy and costly in terms of time and software burden, even for a multitasking computer.

Sinusoidal Input-Output Relationships

An example of the approach is the use of a read-only memory that has the values of $sin\theta$ stored in it for $0° \leqq \theta \leqq 90°$. Two additional digits provide quadrant information, one to complement the input in the even-numbered quadrants, the other to provide the output sign-change for the 3rd and 4th quadrants. The input arrives from an angle-to-digital transducer, the corresponding sinusoidal number values are developed and applied to a D/A converter, and it in turn makes the sine function available as a voltage. If the D/A converter is a multiplying type, computations of the form $Rsin\theta$ are readily performed.



5.4 TRIGONOMETRIC APPLICATIONS

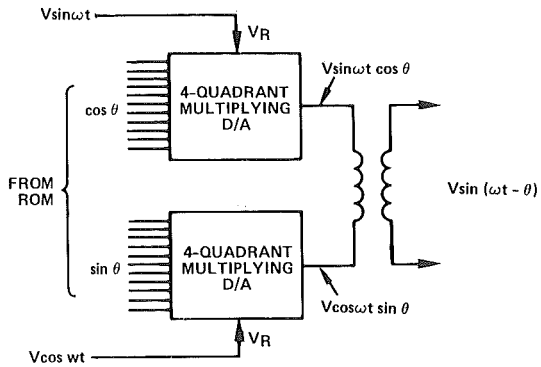
Digital Phase Shifter

The Figure shows two multiplying D/A converters used as digitally controlled attenuators multiplying the reference signals $V\ \sin\omega t$ and $V\ \cos\omega t$ by the vector components of θ. The difference of the outputs of the two converters is then the quantity $V\ \sin(\omega t - \theta)$, where the phase angle θ is set by the converter's digital inputs.

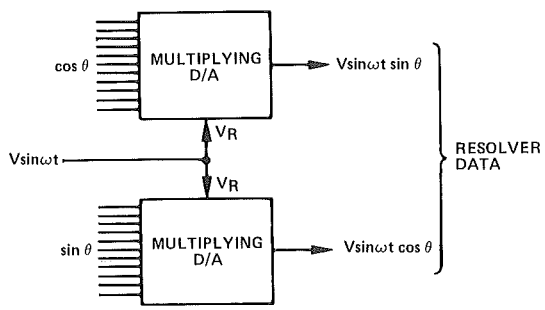
Digital/Resolver Converter (Resolver Simulator)[3]

Similar to the above configuration, but having the common reference input

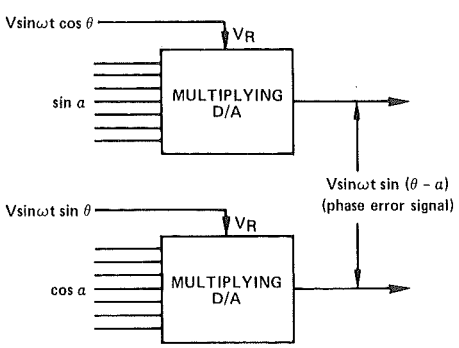
[3]Information on synchro and resolver conversion can be found in condensed form in Chapter 14, and in considerable detail in the book, *Synchro & Resolver Conversion* (1980), available from Analog Devices, Inc.



to both multipliers, V sinωt, this configuration obtains the two components, V sinωt sin θ and V sinωt cosθ, which express resolver data for angle θ. The resolver data can be converted into synchro format with a Scott-T transformer, or an equivalent network in which operational amplifiers provide the appropriate voltage ratios. This resolver simulator can be enclosed within a feedback loop to operate as a resolver-to-digital converter.



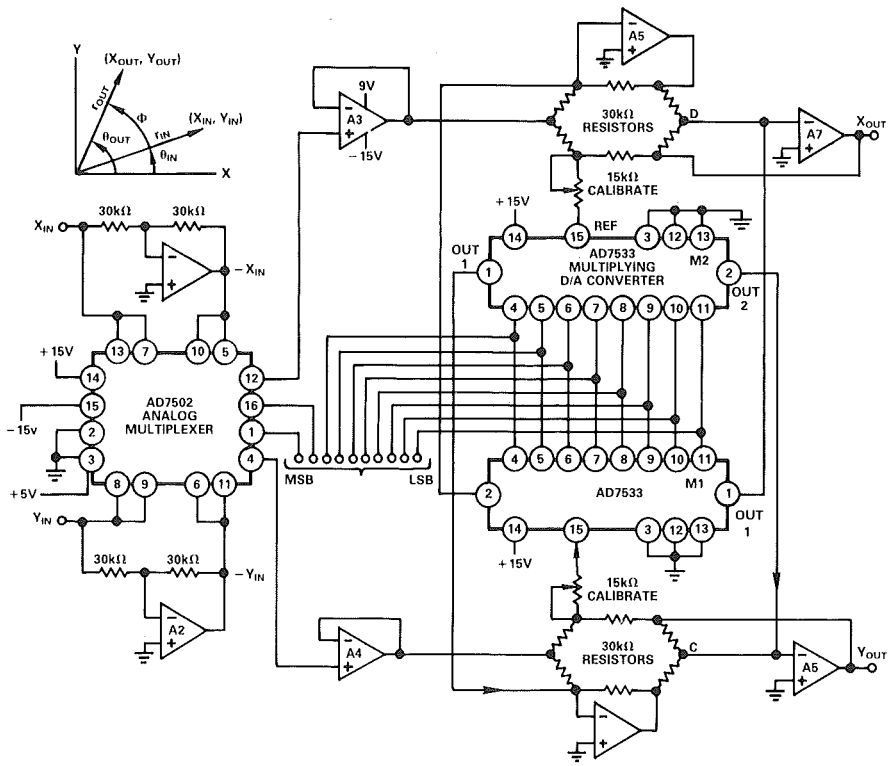
Using the *actual* resolver line voltages V sinωt sinθ and V sinωt cosθ as the converter reference inputs, and multiplying by digital equivalents to cosα and sinα, an output proportional to the angular error, $\theta - \alpha$ (for small angular errors) is developed. Operated in this mode, the configuration simulates a resolver control transformer.



Coordinate Conversion

This interesting scheme takes a vector whose coordinates are X,Y, with angular equivalent $r\underline{/\theta}$, and adds an angle of rotation, ϕ, to produce a vector, $r\underline{/\theta+\phi}$, having a new set of coordinates, X',Y', and the same vector magnitude, with maximum error less than 1°.[4]

It employs two multiplying DACs, a multiplexer to deal with angular inputs in the proper quadrant, a set of precision resistors, and a handful of op amps. The cross-fed summations embody an algorithm that provides a sufficiently good approximation to keep the angle error within 1° and the total harmonic distortion on the order of 1%.



Courtesy of Arthur Mayer, U.S. Patent 3,974,367

5.5 WAVEFORM GENERATION

Linear time functions are generated digitally by clocks and counters, processed by ROM's or μPs for arbitrary wave shapes, and converted to analog functions of time by DACs.[5]

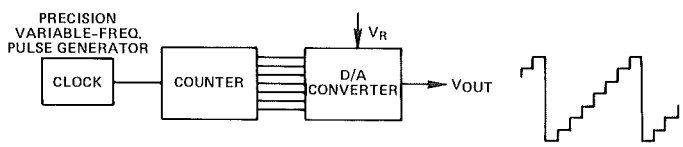
[4]See *Electronic Engineering Times*, July 9, 1979, Arthur Mayer, "Design a Multi-Purpose Network to Rotate Complex Numbers," and *Electronics*, Sept. 22, 1982, A, Mayer, "Low-cost coordinate converter rotates vectors easily," for extended discussions of the circuit, its applications, and PROM correction schemes for eliminating the residual errors.

[5]See also Chapter 6.2.

As long as the original function (with suitable dynamic range) can be created in digital form, then an analog output can be made to follow (within its speed limitations). The ease of manipulation and ability to lock timing operations to precise clocks give the digital approach considerable edge in versatility over many analog alternatives. Deglitching and filtering may be used as (and if) necessary to clean up the waveforms. Variable clock rates or arbitrary counting schedules may be used to obtain staircases having arbitrary, instead of uniform, duty cycles per step.

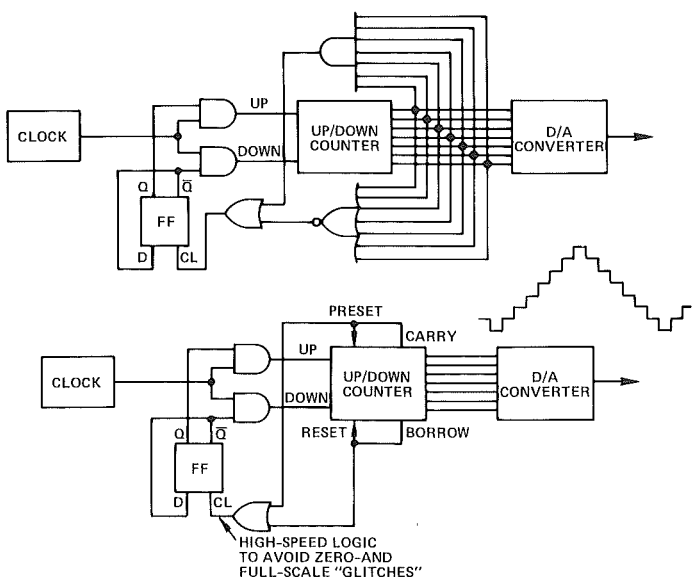
Sawtooth Generator

This sweep generator comprises a digital clock, a counter, and a DAC. The clock pulses increment the counter, and the sequential counter steps increment the DAC output. After the counter is full, it returns to its empty state and starts counting again. Both amplitude and period of the sweep generator are easily and precisely adjustable. The resolution is determined by the number of counts and choice of d/a converter, ranging from the 16-bit AD7546, with its 65,536 steps, down to 10- (or fewer) bit converters with 1,024 steps or less.



Triangular-Wave Generator

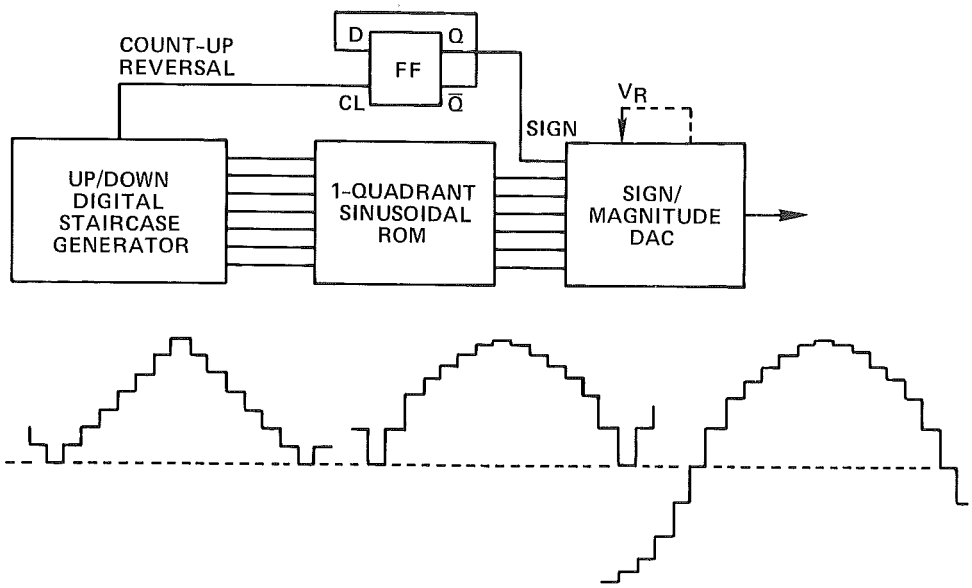
Instead of being allowed to overflow, the counter in this case is an up-down counter that is caused to change direction when it is full and again when it



is empty. Two approaches to reversing direction are shown. In one, the reversal is generated during the full (and empty) states; in the other, it is generated by the carry(borrow) occurring at the leading edge of the next pulse. The result, at the DAC output, is essentially a triangular-wave of precise amplitude and frequency. With little additional logic, full-scale dwell (or dwell-and-reversal at any level) provides trapezoidal waveforms.

Sine-Wave Generator

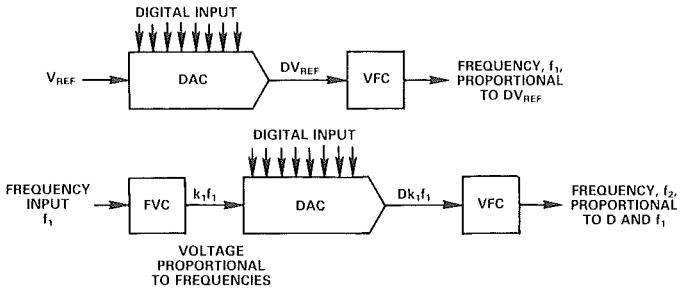
If the digital count is fed to a sinusoidal ROM, and its output, accompanied by polarity information, is applied to a sign-magnitude-coded DAC, the output of the DAC will be an n-bit quantized sine wave. Its frequency is determined by the clock, and amplitude can be controlled at its destination or by the use of a multiplying DAC.



Digital-to-Frequency Conversion

A digital-to-analog converter and an analog voltage- or current-to-frequency converter (VFC) may be combined to generate a pulse train with frequency proportional to the magnitude of a digital word. If the DAC is of the multiplying type, the frequency will be proportional to both the digital input and an analog reference signal, for quadrants in which the VFC receives signals of proper polarity. The excellent resolution of available v/f converters will challenge the highest-resolution DACs.

The frequency of a signal may also be adjusted digitally to arbitrary (including non-integral) values by the use of a frequency-to-voltage converter, a multiplying DAC, and a VFC, as shown.



5.6 TIME RESPONSES

The ability of flip-flops to rapidly acquire and store digital information at rates (and for intervals) depending on precision crystal-controlled clocks, without degradation over time, and the continually decreasing cost of storage capacity, are strong motivations to seek ways of eliminating circuits employing capacitors as storage elements, with their leakage, dielectric hysteresis, parameter drift, and nonlinearity. "Distortionless" time delay, integration, and sample-hold are a few targets for such effort.

Precision Analog Delay Line

There are interesting applications for good analog delay lines: analog correlation, "distortionless" signal compression or expansion (e.g., "riding the gain" without missing a drumbeat), electronic echo-chamber effects, analog modeling of processes that incorporate pure time delay for predictive control, design of filters with arbitrary transfer functions, are a few.[6]

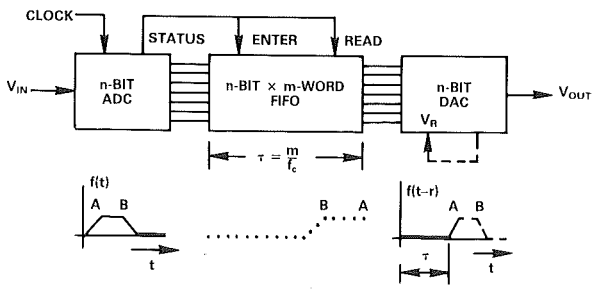
But there hadn't been a decent way of building a practical analog time-delay device that is variable over microseconds to minutes to months, without tying up a processor, until converters became available at low cost and FIFO (first-in, first-out) memories became available with reasonable word widths and stack lengths.

Active or passive filter-type delay lines were seldom "distortionless," analog "bucket brigades" had excessive leakage errors at low speeds, as well as a resolution-vs.-cost problem (this latter being solved by charge-coupled MOS high-speed bucket brigades), tape recording wasn't efficacious at high speeds, and the use of mainframe memory was too expensive (and bulky for portable instruments).

In the example shown, the delay is produced by FIFO registers (e.g., 9 bits $\times$ 128 stages). As the word resulting from each conversion is clocked into the FIFO, the earlier words are advanced; after 128 clock pulses, each input word emerges from the output port, with a delay of 128 clock pulses. For 9-bit conversion, signals that can be quantized into 512 discrete levels can be delayed

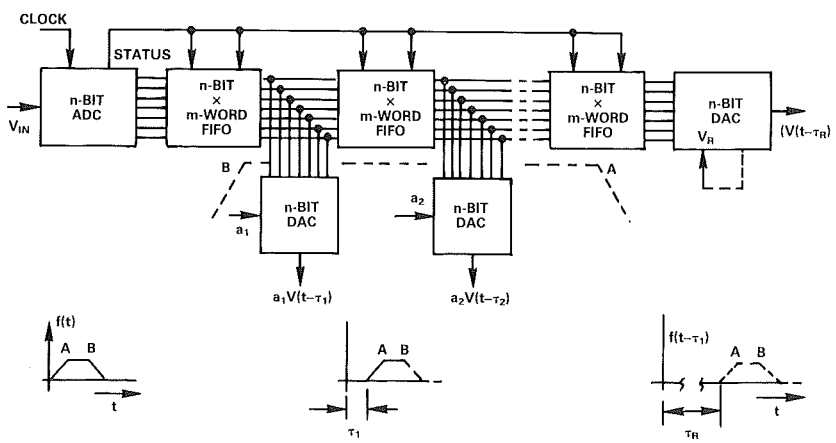
[6]See also Section 6.2.

with a resolution of 1/128 of the delay time (e.g., 2μs of 256μs, 1s of 128s, or 2.81° of a sinusoidal ac signal of period equal to the delay time, etc.). FIFOs can be stacked in parallel for increased bit-resolution, and in cascade for increased time-resolution.



Tapped Delay Line

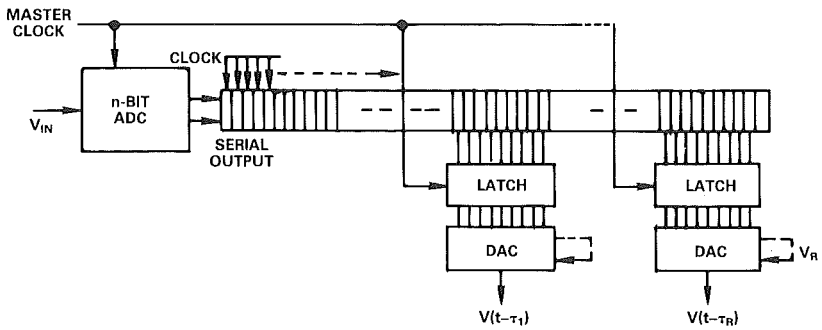
This device makes a number of points in the history of a waveform available simultaneously. It is simply the delay line with an increased number of discrete "chunks" of delay, and readout via DACs at each point. Multiplying



DACs allow such interesting functions as $f(t) \cdot f(t-\tau)$ to be computed for a variety of values of τ. Hybrid IIR (infinite impulse response) and FIR (finite impulse response) filters with predictable indicial time responses may be constructed by this technique.

Serial Delay Line

For signals that do not require sampling at top speed, a considerable saving of the cost of FIFOs (or increase in the time-resolution of the delay) can be achieved by feeding the converted signal into the line serially, using shift registers for delay, and converting back to parallel information for the D/A conversion. Required shift register capacity depends on clock rate vs. delay, delay resolution, number of pickoff points, and length of data words.

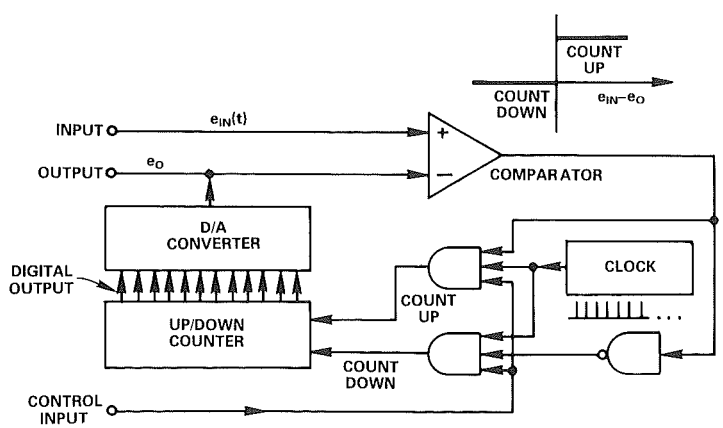


5.7 DIGITAL SERVO CIRCUITS

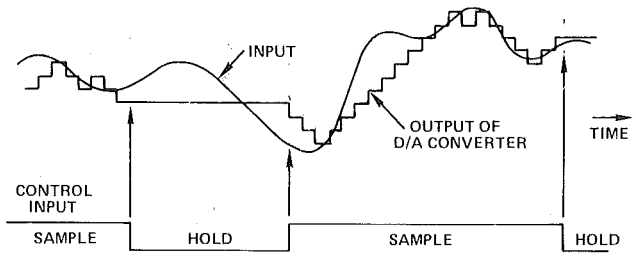
Most a/d converter designs involve feedback. Thus the very means of conversion implies that the combination of analog-digital interaction and the power of feedback can yield quite valuable results. A few examples are sample-hold, peak-detecting, and automatic zero-setting.

Tracking Sample-Hold (A/D Converter)

This circuit, also mentioned in the chapter on sample-holds, is especially useful as a track-and-infinite-hold device. It can acquire the analog signal within a minimum of 1 count and maximum of 2^{n-1} counts, and, upon command, *hold* it indefinitely without degradation, providing both digital and analog readout. Since it uses an up-down counter, it will track the analog signal at a constant rate (2^{-n} FS per count), and "hunt" between the two digital values that straddle the analog value, if it remains constant. Hunting may be avoided by the use of hysteresis, but a lag will be introduced.



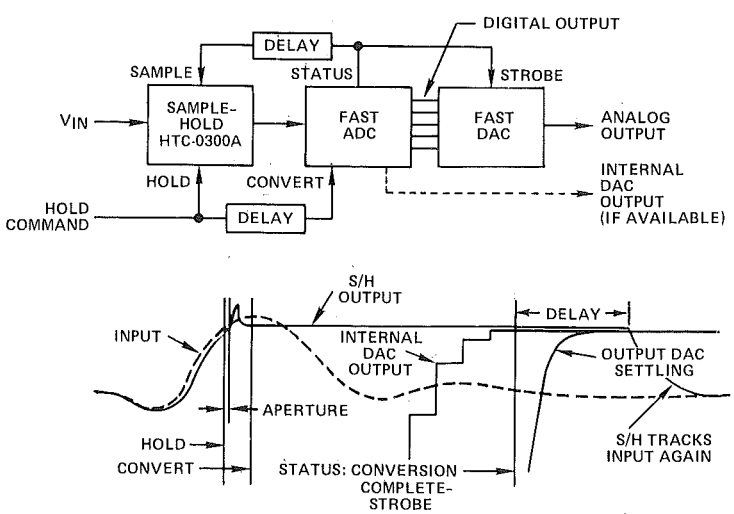
"1": SAMPLE, "0": HOLD

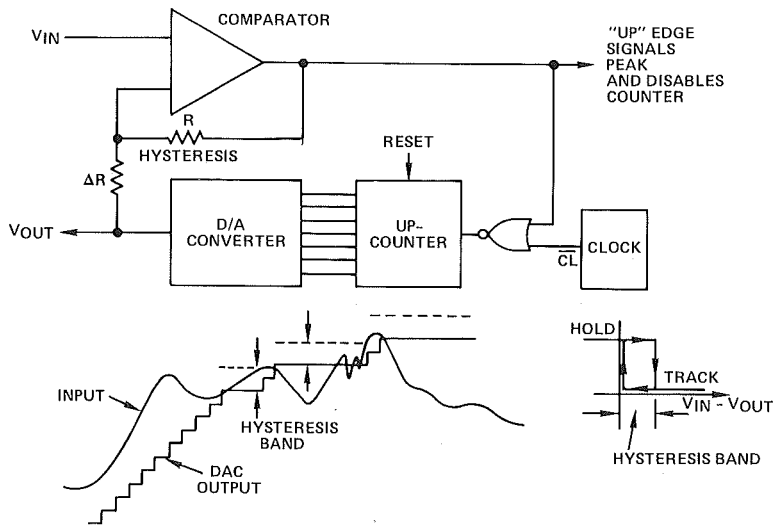


For analog signals without the sharp changes in slope, the tracking a/d converter is one of the lowest-cost ways to convert, since it eliminates the need for a sample-hold. However, its conversion time is variable, which introduces timing errors in sampled-data systems, since the most-recently acquired value may represent any value of signal during the interval between interrogations. Also, its response depends to a great extent on the amount and type of noise present. Tracking converters are discussed in Chapter 14.

Digital Pulse Stretcher

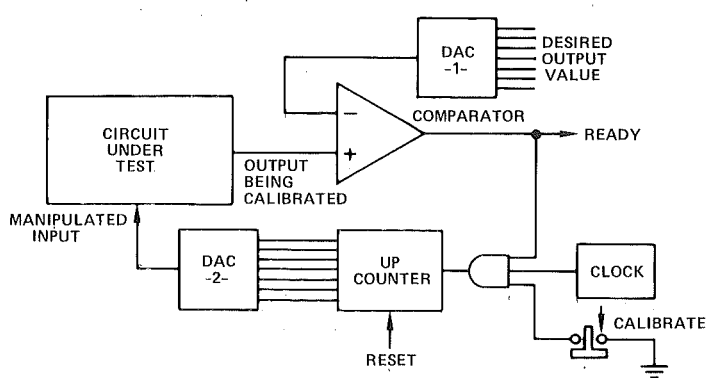
For extremely fast-acquisition-very-long hold, this circuit, consisting of a fast sample-hold and a fast successive-approximation A/D converter will provide the best results. Both analog and digital outputs are available. For single samples, if the internal D/A converter's output can be made available without slowing conversion, the output D/A shown in the Figure is unnecessary. The HTC-0300A is kept in *sample* at all times except during conversion. When switched to *hold*, it should have a "head start" of 200ns (acquisition time) for its transients to die down before the first conversion decision is made. Aperture time is about 6ns with 100-ps jitter.





Digital Peak-Follower (with Hysteresis)

Similar to the tracking sample-hold, but using an up-counter (a valley-follower would use a down-counter), this circuit will *hold* the highest value of input that it has been able to track. However, to provide a small measure of immunity to noise, hysteresis makes the circuit insensitive to small changes; in order for the input to be followed, it must be higher than the stored value by a preset amount. A similar circuit can be used for valley following, and two such circuits with digital or analog subtraction will provide peak-to-peak measurement. As with all circuits employing comparators, care should be taken to avoid oscillations; ΔR is a very small fraction of R.



Automatic Set-Point Circuit

If a circuit under test is to be calibrated from time to time (e.g., each time some element, perhaps a device under test, is changed), the resetting and the level to which a test value must be reset, may be adjusted digitally. In the example, an output of the circuit must be set to a value equal to a calibrating value set by high-accuracy standard, DAC-1. The values are compared, and

a clock increments a counter, which updates a DAC, setting the input that performs the calibrating adjustment. When the comparator changes sign, calibration is complete, and the sign change indicates a "Ready" condition. The calibration value is retained until a new calibration cycle is initiated by resetting the counter and gating the clock.

5.8 A FINAL NOTE:

SOFTWARE vs. HARDWARE

The examples given here all involve hard-wired analog-digital circuitry. For applications in which microprocessors are available, it should be evident that functions involving memory, control logic, and digital data can be as well (and more flexibly if perhaps not quite as speedily) handled through the writing of appropriate microprocessor programs, at the cost of software development—and program running time.

Rather than viewing the techniques as competitive, the designer should consider that the three-way tradeoff between analog, hard-wired, and software approaches provides at least one more degree of freedom for developing cost-effective instruments, apparatus, and systems. The flexible designer will not arbitrarily exclude a given approach; on the other hand, the committed designer should know that other alternatives to one's own predilections do exist and may, on occasion, prove available to save the day.

Chapter Six

Applications of Converters in Instruments and Systems

Chapters 1-5 have introduced the basic hardware elements of systems and equipment that involve converters, shown the basic configurations of data-acquisition and data-distribution systems, and indicated a few examples of the uses of digital and analog elements in intimate combination.

In the real world, converters are necessary whenever data in analog form must be processed digitally—or devices that require analog inputs must get their input data from digital sources. Thus, any instrument, apparatus, equipment, or system involving real-world data is a potential home for one or more data converters.

This chapter will illustrate a few examples among the plethora of systems and equipment that have been conceived of or built involving converters. The examples are drawn from a variety of sources, but they share the ideas, hardware, and circuit structures that have already been touched upon.

The intent is to inform the reader of what has been done, to suggest what can be done, and to arouse thoughts of what *might* be done by adding the conceptual tools described in this volume to the fund of knowledge and experience already existing pertaining to the reader's own field of endeavor. To the digital expert, it should provide insights into the real-world connection; to the analog expert, it may suggest ways of doing analog jobs better with digital assistance; and to the person with much theory and little practice, it should give a more-

concrete feeling for practical applications of digital techniques in the real world.

6.1 AUTOMATIC TESTING

> "Automatic testing of electronic devices has been a major factor not only in the overall improvement of product quality and reliability, but also in the dramatic lowering of product costs."[1]
>
> – Harold T. McAleer, General Radio Company

Although electronic devices are a major (and in some ways an obvious) market for electronic testing equipment, their makers and users are by no means the sole beneficiaries of automatic testing. Anyone whose blood has been tested recently, has flown safely in a modern jet aircraft, or has an automobile that has been inspected with modern equipment, has been exposed to the potential savings (and not just financial) inherent in automatic testing.

The cost savings, both immediate and long term, result from a number of characteristics of automatic testing:

Human resources are conserved. Fewer persons can conduct more (and more-thorough) tests of high complexity with minimal training.

Volume. Large numbers of tests can be performed in a short time, including individualized tests on complex devices and repetitive tests on large numbers of simple devices.

Reliability and consistency. A well-designed test program will perform identical tests leading to consistent results, with no aberrations due to misreading, fatigue, etc. If failure occurs in mid-test and repairs are made, the entire test cycle can be repeated, numerous times if necessary, with full confidence that the most recent test has "cut no corners."

Multiplexing of adjustments and readouts. An instrument designed for use in automatic testing bears little physical resemblance to conventional instruments, since it need have neither binding posts, knobs, readout, nor even "front panel;" it shares the system's readout devices; connections and adjustments are made by the system.

Automatic Calibration. Any necessary calibrations, zero adjustments, nonlinear-device compensations, or other predicted allowances, whether of the test subject, the sensors, or the test system itself, can be made under system command. Adaptive range-changing can be fully automated.

Measurement statistics. The system can retain in memory the results of all tests, the results of discrepant tests, and data on specific parameters; it can number-crunch the statistics and print the results upon request. Yield studies can lead to product improvements, elimination of sources of repeated rejections, and prediction or tracing of future failures.

[1] *IEEE Spectrum*, May 1971, "A Look at Automatic Testing"

In short, a well thought-out, well-designed, and well-implemented automated test facility can reliably perform large numbers of tests, around the clock, on a "100%" basis, consistently and without tiring, with accuracy and skill, and with feedback to the designer for the next generation of the product. Skilled test personnel can be freed for more-creative pursuits because they don't have to follow long, detailed procedures for routine manual testing of the ins-and-outs of complex systems, calibrate instrument dials, interpret go-no-go limits, and calculate odd ranges.

Then there are the important but less-measurable results, that pay off in human values as well as dollars-and-cents: the aircraft engine that didn't fail, the electrical chassis that didn't need field repair, the steel rolling mill that didn't run away, the hospital patient that survived, the vendor whose reputation remained consistently high.

6.1.1 USES FOR AUTOMATIC TESTING

The manufacturer of components, such as integrated circuits, benefits greatly, because testing is a far-from-negligible cost in the integrated circuits business. Besides delivering a higher level of acceptable quality to the customer, the manufacturer also develops more-accurate knowledge of yields and trends, and can develop specific selection categories for special orders. For the producer of high-performance specialty ICs, such as Analog Devices, it is an indispensable tool. Such devices as laser-trimmed low-offset op amps, high-accuracy monolithic multiplier/dividers, and—indeed—full-accuracy converters, with resolutions of 12 or more bits, would be so costly as to be infeasible if manual measurements and adjustments were involved. (Instead, the additional cost is a small fraction of the price of the untrimmed unit.)

The user of large numbers of identical components can also benefit: Machines can be used to weed out discrepant units in incoming inspection; measure, select, and grade units for different applications (freeing the user from paying the manufacturer extra to do the same job) and keep comparative statistics from lot-to-lot and vendor-to-vendor. It may be noted, as a matter of perspective, that an average saving of 10¢ on 100,000 units is $10,000.

The manufacturer of equipment and systems can test subassemblies in-process, or as received from subcontractors; (s)he can also test completed pieces of equipment thoroughly. In both cases, the test system can be programmed for GO/NO at points of discrepancy, and to either reject the device for later evaluation, or branch into a diagnostic mode, to isolate the portion of the circuit (or perhaps even the component or connection) that is faulty. Repaired units can be recycled and subjected to the same battery of tests as the new units, and serialized records of all such processing can be maintained.

Highly complex systems, such as jet aircraft and their various subsystems, can be tested thoroughly on the ground by a small number of persons in a short

time, with a high probability of finding any faults, or the discrepancies that might indicate incipient faults. In addition, the on-board test and monitoring system can provide warning to the crew (using appropriate media) of anomalous subsystem behavior, provide automatic switching to backup systems, and, as the electronic portions become increasingly sophisticated, it can perform a degree of diagnostic testing, facilitating repair.

6.1.2 INGREDIENTS OF TEST SYSTEMS

For systems that test devices, the test begins with the *unit under test*. It must be handled, maneuvered into place, and connected to. Then a *stimulus* is applied, and a *response* must be measured. The response is compared with a set of possible responses, and a *decision* is made (accept, reject, grade-and-sort, perform an adjustment) and communicated (print, store, mark, analyze), and a new instruction is given (next test, next set of connections, next device, wait for manual instruction, etc.). An outline of such a system is shown in the block diagram of Figure 6.1.

The simplest devices have two leads (resistors, capacitors, diodes), but most devices or equipment subject to testing will have many more. For example, ICs often have more than 20 or 30 connection pads or pins, and printed-circuit boards subjected to all-node "bed-of-nails" testing may have more than 1,200 connection points. The program must call for connecting the appropriate

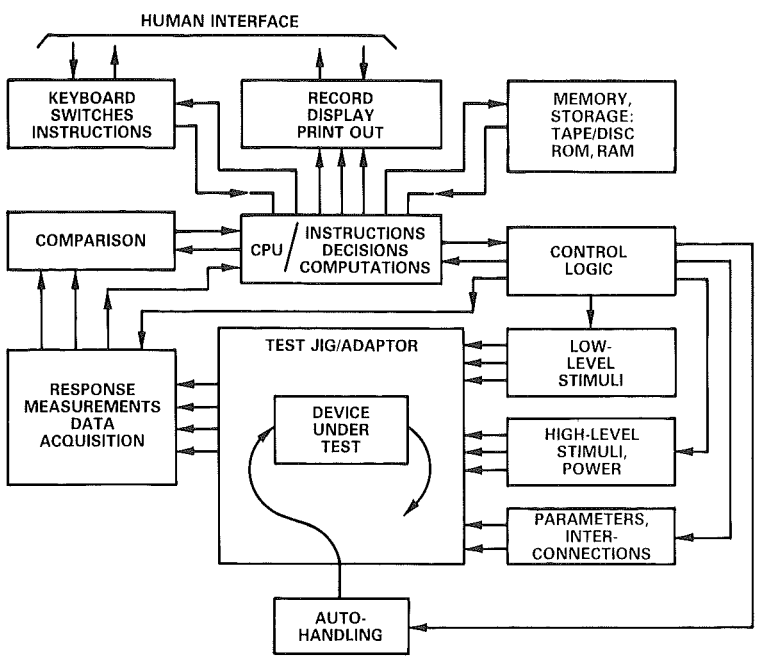


Figure 6.1. Test system ingredients in typical configuration.

stimulus generators and power sources to the appropriate terminals, and the appropriate measuring devices (bridges, amplifiers, etc.) to *their* appropriate terminals, and making all opens, shorts, "grounds," links, etc., as required for the test step. Some of these may be hard-wired in the adaptor; others must be called for by software or operator setting. It is absolutely essential that noise pickup and interference, as well as parasitic effects caused by lead resistance, capacitance, and inductance, be minimized.

A typical flexible, modular architecture employed in general-purpose electronic component and IC testers is shown in Figure 6.2. The device under test is plugged into an interchangeable socket board wired appropriately for the specific type. The socket connects to a socket assembly, wired for specified conditions; it in turn connects to a pluggable family board, which provides stimuli, gains, and special functions needed to exercise and test a particular group or class of devices (e.g., op amps, a/d converters, d/a converters, digital logic ICs, etc.)

Inside the system housing, the measurement section, which comprises the measurement card, source card, and digital I/O card, provides program-controlled measurement functions, voltage sources and references, and the digital I/O drivers and detectors required for performing device tests. It operates from a test-system bus, controlled by a 16-bit CPU. Besides performing the

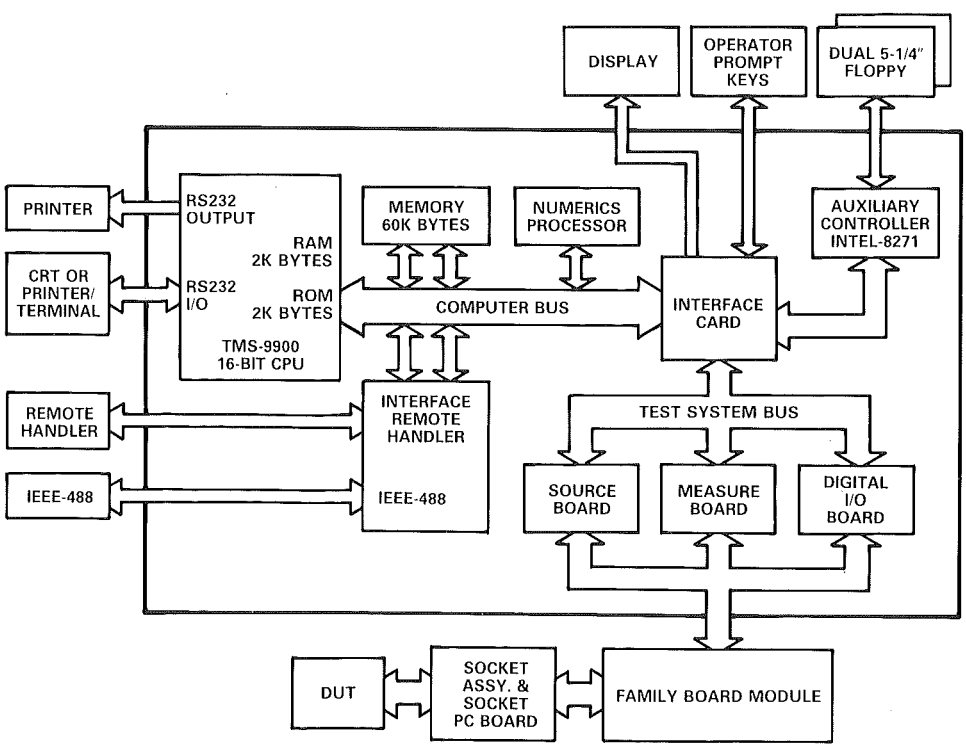


Figure 6.2. Block diagram of Analog Devices LTS-2015 device test system.

actual tests, the system interfaces with remote automatic handling devices, printers, and CRT terminals, and communicates locally via an alphanumeric keypad, operator prompt keys, a display, and one or two floppy-disk memories.

Converters in Test Systems

It may be fairly evident that much of the engineering and hardware cost of test systems goes into fixtures, switching devices, computers, peripherals, displays, wiring, and cabinetry. However, since the analog stimuli—and comparison-type measurements—are controlled digitally, and the analog responses must be returned to digital form for processing, it should be evident that converters and their accessories play a key role in ensuring test accuracy, speed, and reliability, yet represent but a small fraction of the cost of the system. For this reason, it may be false economy to use conversion devices that are anything but entirely adequate to do the job, or to seek to cut cost corners by risking marginal performance.

In the system shown in Figure 6.2, the reference system consists of a traceable temperature-controlled precision voltage reference and a 16-bit-accuracy 12-bit-resolution d/a converter. Figure 6.3 shows how DACs are used in the measurement section of that tester. Note the use of force-sense (i.e., Kelvin) connections for both the amplifier output and the ground return to enforce accurate test voltages at the device under test, irrespective of line drops.

Typical uses of D/A converters in testing include: programmable power supplies, pulse generators, sweep generators, waveform generators (with appropriate digital inputs). They may be used as offset and gain "potentiometers" in calibration loops, as bridge-balancing voltage sources, and as part of A/D converters, sample-holds, peak-followers, etc.

A/D converters, either with multiplexing or per-channel, return the measurements to digital form, often after processing by isolation or instrumentation amplifiers, by op amps as electrometers, and, in some cases, by multipliers, ratio devices, log devices, and all the other paraphernalia mentioned in Chapter 2.

An essential decision that must be made is the degree to which analog data reduction and/or digital signal-processing hardware will be used, as compared with the performance of similar functions by digital software. This consideration depends on such factors as the amount of number-crunching necessary and the time available for it, specified accuracies, cost and energy tradeoffs, as well as the background, experience, and inclinations of the designer. We suggest that analog-oriented designers not overlook the possibilities of software and digital signal-processing hardware for reliable routine computation, and that digital designers consider the decreasing cost of functions that can be performed with analog modules and linear integrated circuits at the front end, and the balance between too much and too little data.

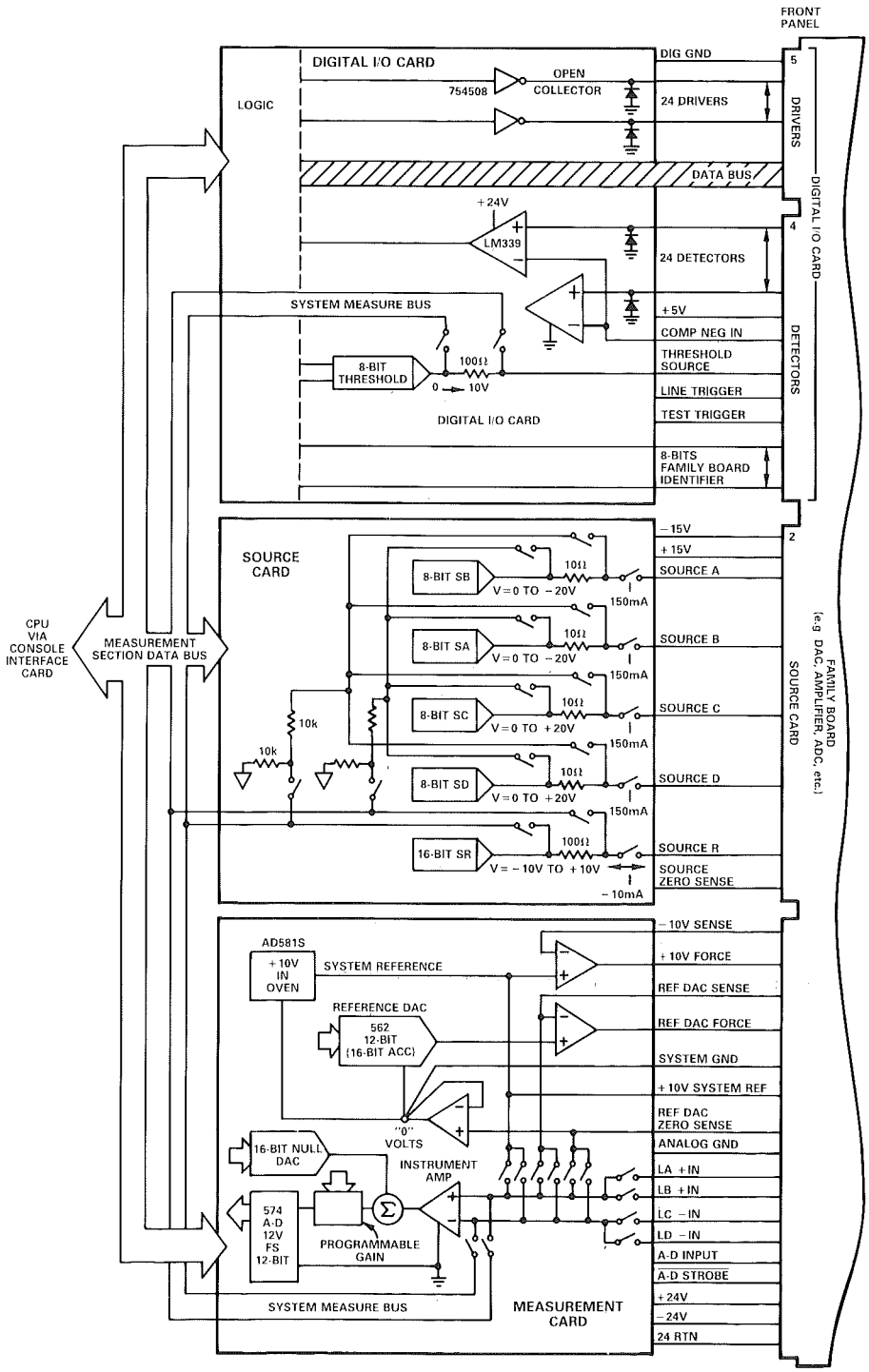


Figure 6.3. Measurement section of LTS-2000 tester, showing measurement, source, and digital I/O cards, with their connections to the family board.

Much that could be said about system optimization, in terms of getting the best-possible interference-free measurements of suitable accuracy, has already been mentioned in several places in this book (for example, Chapters 12 and 22); and test systems are probably the most representative class of design problems requiring active application of these principles. In general, it is best to keep the conversion function as close as possible to the analog measurements, especially in applications involving remote handlers.

In component tests, where the lead-runs to the unit-under-test (UUT) are controllable, as is the local environment, the main sources of interference arise from the proximity of input, output, power, and logic leads in the vicinity of the test adaptor. In large-system testing, long leads, including multiconductor cables and connectors; the presence of electrical noise (RFI, power line and switching-transient spikes); and possibly unfavorable environmental conditions (temperature, humidity, vibration), may combine to make the measurement problem extremely difficult.

It often turns out that, in the design of large systems, performing tests locally is an effective way of solving the interference problem, using a local μP-controlled self-contained measurement subsystem (or *instrument*) that communicates digitally with other such subsystems under the direction of an external system controller. The IEEE-488 bus structure was designed specifically for communication of data, instructions, and handshaking in tests involving sources (e.g., signal generators), measuring devices (e.g., precision voltmeters), display devices (e.g., printers), and controllers (e.g., keyboard/display terminals).

6.2 DIGITAL SIGNAL PROCESSING

In this section, we shall discuss briefly the class of digital applications that involve the generation, transmission, delay, recovery, processing, storage, characterization, and synthesis of analog waveforms. Conceivable applications include:

Time expansion, compression, (relative) advance, and delay
Transient storage and recording
Synthesis and analysis of speech and music (and waveforms in general)
Transfer-function synthesis and analysis
Convolution
Digital filtering
Recovery of signals from noise by correlation techniques and fast Fourier transforms
Scrambling and unscrambling of coded transmissions
Generation of arbitrary signals and transfer functions

Digital methods, especially with microprocessors and digital signal-proc-

essing chips, can provide a powerful set of tools for dealing with analog functions and the transfer functions that are used to shape them in the time and frequency domains, as we have suggested in Chapter 5. These methods, and some components that make them feasible in a reasonable time frame are discussed briefly in Chapter 21 and at greater length elsewhere.

The key that unlocks the door is the A/D converter, which "freezes" a sample of the waveform and makes possible permanent storage without degradation. Thereafter, digital shift registers, multipliers and multiplier-accumulators, memories, comparators, microprocessors, and control logic can perform a virtually unlimited set of operations digitally at any time (*on-line* or *off-line*).

Errors are due to the discrete-time and quantized-amplitude nature of the sampled signal, and truncation or roundoff errors in computation (where necessary or permitted). If sampling occurs at an adequate rate, if the conversion has sufficient resolution, and if the computation carries enough significant bits, there is no loss of information, even though the signal be stored, multiplied, integrated, added, subtracted, correlated, or otherwise man(or machine-)ipulated. Via d/a conversion, the data can of course be returned to the analog domain and subjected to further processing there, but its attributes can be retained in digital memory for as long as desired.

Of the circuits and ideas that appear here, some are variations on the basic theme of the delay line, others are applications of basic digital signal processing techniques; they represent promising areas of application but are not necessarily new or original. Their purpose is to unleash the reader's curiosity and creativity, in the field broadly encompassed by the title of this section. We've tried to avoid, except where necessary, mathematical particularities (and the controversies they sometimes engender), since the purpose of this chapter is only to relate converters to the more interesting applications they are used in. The field is large enough that a book the size of this volume would be required to deal satisfactorily with it.

6.2.1 SHIFT-REGISTER DELAY LINE

The basic tool for performing many interesting functions is the shift-register delay line—with and without taps—mentioned briefly in Section 5.6, and shown here again (Figure 6.4) for further discussion. It should be noted that the delay line need not be an isolatable physical entity—it may be a sequence of memory locations in a microcomputer, incremented or decremented on appropriate clock pulses.

Suppose the analog signal is a one-shot occurrence, of which m samples have been taken, the sequencing clock has stopped, and the conversions have ceased. The signal is now stored in the delay line in digital form, and it will remain there until it is advanced or cleared or the power has been turned off. A number of interesting things may be done with the stored signal:

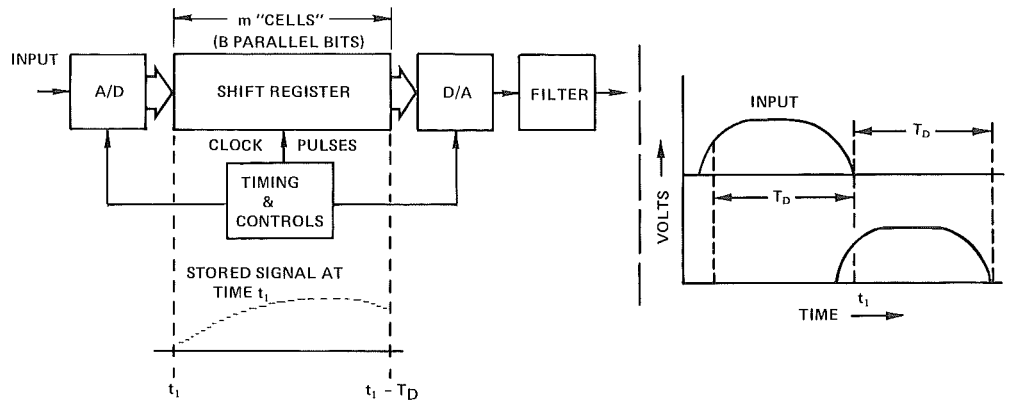


Figure 6.4. Digital delay line (word- or byte-wide shift register).

Read out into memory

The delay line can be considered as FIFO (first-in, first-out) buffer memory. The stored signal can be read out of the delay line, a word at a time, and stored elsewhere in memory, while the line awaits another transient (Figure 6.5).

Readout as an analog signal

The signal can be read out without further processing and converted to an analog signal with a DAC, each sample in turn—but at an arbitrary rate, determined by the choice of clock frequency. For example, the transient may have been quite rapid, but it is desired to plot it out on a chart recorder. Or, it may have been fed into the line slowly (perhaps even keyed in manually and asynchronously as an arbitrary waveform), to be used as a shaped stimulus for an analog process, and it is to be discharged at high speed. In applications such as this, digital techniques have a distinct advantage over purely analog switched-capacitor "bucket-brigade" delay lines, because there is no loss of accuracy with storage time or cumulative number of sections, hence no resulting constraint on either parameter.

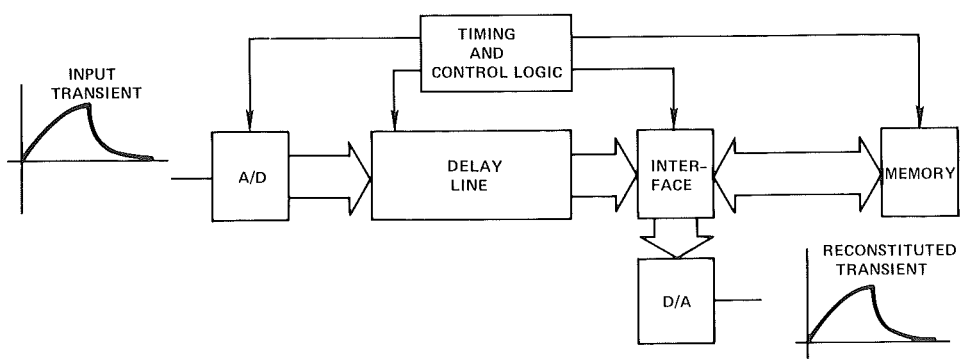


Figure 6.5. Digital delay line as FIFO buffer (transient recorder).

Recirculate

If the the data stored in a delay line is fed back repetitively from the output to the input end of the line (or the first of the memory locations), it becomes a recirculating delay line (Figure 6.6). The stored signal will then appear at the end of the line cyclically, allowing the signal to be displayed on an ordinary oscilloscope. By loading, or "charging", with an arbitrary or an analytic input signal (derived from either an analog or a digital variable), then providing rapid recirculation and D/A conversion, it is possible to create an extremely wide range of arbitrary repetitive analog waveforms, of controllable repetition rate and amplitude.

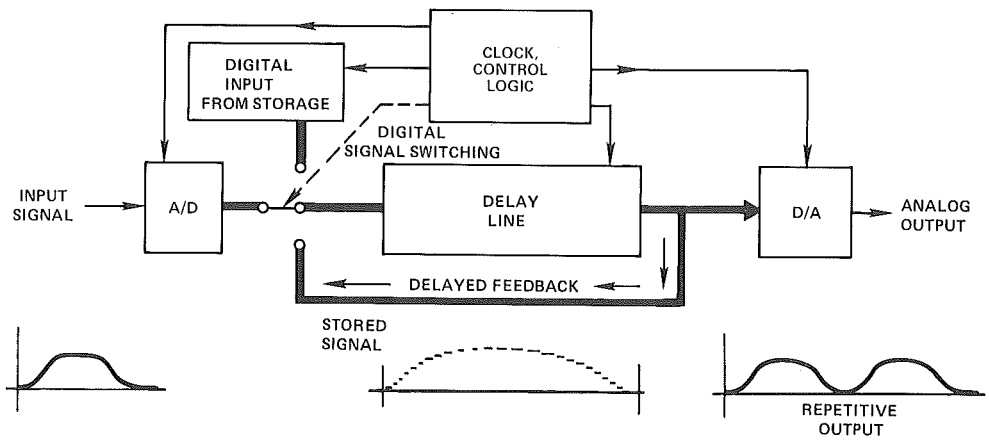


Figure 6.6. Recirculating delay line.

Perform waveform averaging by addition

If the same message is sent repeatedly but arrives at the converter accompanied by (and perhaps "buried in") noise, it can be recovered by summing all the versions of the message synchronously: the coherent portions will add directly with the number of items summed, while the rms noise will tend to be "averaged out" and will increase only as the square-root of the number of items. For example, with 100 repetitions, the signal will be increased in relation to noise by a factor of 10. This can be accomplished with a delay line or set of memory locations by summing the sample increment of the newest message in each position with the sum of the corresponding samples of previous messages, accumulated in the delay line. Thus, when the second message arrives, its first-position sample is summed with the already-stored sum of the previous first-position samples; its second-position sample is summed with the sum of the previous second-position samples; and so forth. Since the original messages are presumably identical, while the noise varies randomly, each iteration adds 1 unit of original signal to each position, while the noise components tend to be averaged out.

In practice, computing the simple sum

$$Y_{r,n} = \sum_{i=1}^{n-1} X_{r,i} + X_{r,n} \tag{6.1}$$

– where $Y_{r,n}$ = sum of n samples at the rth position, Σ = output of the delay line (n-1st sample at the rth position), and $X_{r,n}$ = input of the nth sample at the rth position—and averaging it by dividing by n afterwards leads to an open-ended stored-signal amplitude, which requires a large dynamic range (wide delay line). Thus some form of normalizing is desirable.

Figure 6.7 shows one scheme, based on the algorithm of (6.2):

$$Y_{r,n} = Y_{r,n-1} + \frac{X_{r,n} - Y_{r,n-1}}{n} \tag{6.2}$$

where $Y_{r,n-1}$ is the delay line's output (n – 1st sample at the rth position).

At the rth position, this is equivalent to adding the 1/n of the new input to the previous output, multiplied by (n – 1)/n. Thus, for n = 1, the output, $Y_{r,1}$, will be equal to $X_{r,1}$; for n = 2, the output will be equal to $(X_{r,1})/2 + (X_{r,2})/2$; for n = 3, the output will be equal to $(2/3)(X_{r,1}+X_{r,2})+(1/3)X_{r,3}$, and—in general—the output is equal to the average over n inputs; The average value would be constant for n equal input values at position r. Since all variables are normalized, it is not necessary to carry large summation values or perform more than one division.

It is interesting to note that, as n becomes large, for identical inputs, Y_r, $Y_{r,n}$ becomes very nearly equal to $Y_{r,n-1}$, because each additional increment causes little change, being divided by n. For fast results, 1/n can be obtained from a lookup table[2] and multiplied by the difference in a digital multiplier-accumulator, which takes the product and sum.

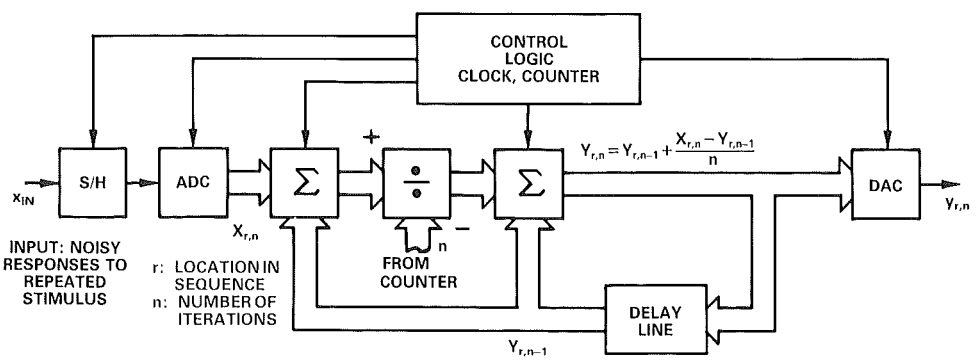


Figure 6.7. Waveform averaging scheme, in block-diagram form.

[2]The size of the lookup table and speed of computation can be minimized by the use of an approximation technique. See "Fast, Simple Approximation of Functions," by Matt Johnson, *Analog Dialogue* 18-1 (1984).

Time Compression by Sampling

This application employs a stroboscopic effect to apparently speed up a slowly varying signal.

In Figure 6.8, a shift register (or set of memory locations), is incremented at a high frequency, f_c, for example 513kHz. The converter is digitizing a slowly-varying signal at a rate f_s. Suppose that: the shift register has 512 steps, the line is full of previous values, and at a given instant, the 512th sample appears at the output and is fed back to the input. On the next step, starting the mth iteration, the a/d converter output, $X_{1,m}$, is fed onto the input bus to replace the existing value, its previous output, $X_{1,m-1}$. The line then advances for 512 steps. On the 512th step, the input is once again $X_{1,m}$; and $X_{2,m-1}$ appears at the end of the line, while $X_{2,m}$ is ready at the converter output. On the 513th step, the converter output is fed into the line to replace $X_{2,m-1}$. $X_{1,m}$ and $X_{2,m}$ are now indexed down the line, and on the 513th step, $X_{3,m}$ replaces $X_{3,m-1}$. By the time 512 conversions have occurred, in real time, each consecutive sampled signal (including new and previous values) has circulated 513 times, thus providing a 512-fold speeded-up version of the (for example, 1.96Hz) analog input waveform at the output of the D/A converter, at the equivalent of 1ms per sample.

If each cycle of the analog waveform is identical to the adjacent ones, and if the clock is synchronized to the analog signal, the output of the DAC, plotted on an oscilloscope screen, swept at 1kHz, will appear to stand still, plotting the low-frequency input, but *with no flicker*. Changes to the input signal, from iteration to iteration, will appear as progressively appearing changes to the stationary pattern. Since the compression ratio depends on the time required for each 511 samples, it is proportional to the clock frequency, which can be locked in at any convenient value.

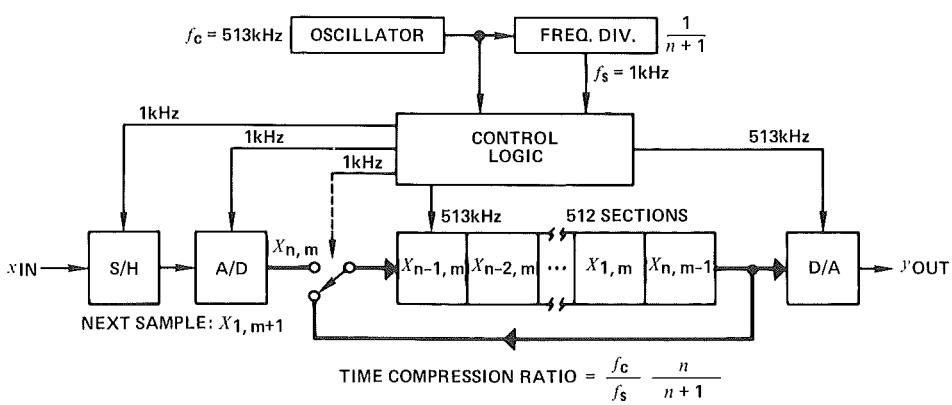


Figure 6.8. Time compression scheme, using a circulating delay line.

Real-Time Correlation

For an input function, f(t), the output of the delay line (or set of memory locations) over a complete circulation (in compressed time) is a set of values of $f(t - \tau_i)$. If the successive values are multiplied by the sampled value of another waveform, g(*t*), which, with f(*t*), is updated after each circulation, and if each individual product is averaged with its synchronous counterparts from previous circulations, as described earlier, the output of the averager will represent a sample-by-sample cross-correlation of *f* and *g* at a real-time rate, delayed by the product of the sampling period and the number of samples circulated (Figure 6.9).

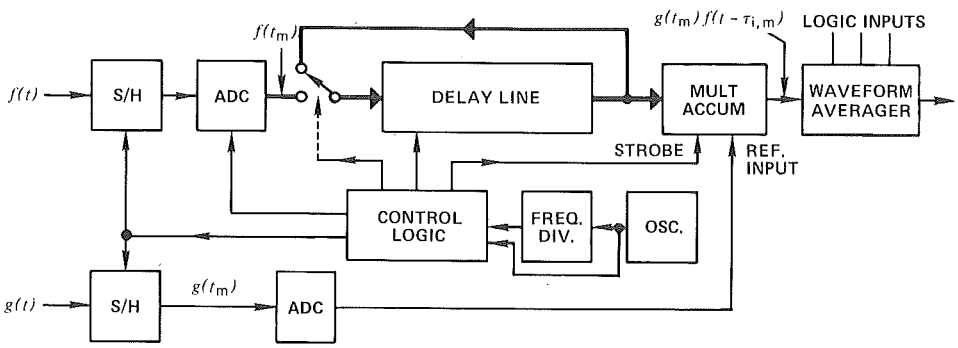


Figure 6.9. Real-time correlation scheme, employing a recirculating delay line and a waveform averager.

Incremental Delay Line as a Filter

If the delay line consists of a number of sections (or successively incremented memory elements), and the outputs at the taps are multiplied by arbitrary coefficients and summed (Figure 6.10), it is possible to synthesize arbitrary

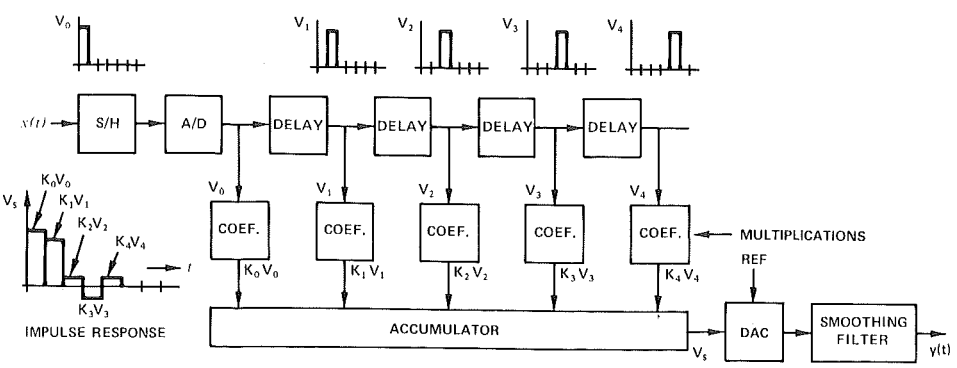


Figure 6.10. Delay line as a non-recursive filter with programmable real-time response (small number of sections shown for clarity). If fastest possible speed not necessary, fewer hardware multipliers are required; a single multiplier-accumulator could be adequate.

time-domain responses to steps, pulses, or other waveforms. Since the output bears a linear* relationship to the input, the resulting transfer function may provide amplitude and phase responses to other signal forms (over limited ranges of frequency) that can be expressed by transform integrals but were once otherwise formally considered "unrealizable."

In this case, a *non-recursive filter*, the output is a function of the input only and is inherently stable. In practice, if the signal is not too fast, a single multiplier-accumulator may be programmed to perform all the multiplications and the summation digitally, followed by a single output DAC, saving hardware. If the filter's response is defined in terms of frequency response, instead of time-domain response, the coefficients—although messy to compute from scratch—are handily computed by cut-and-dried techniques, such as the Remez Exchange Algorithm[3].

Recursive Filtering

When the output is a function of input only, the number of possible responses is limited, because—barring the introduction of recursion via an external feedback system—the output will settle within a finite time after the input has ceased to vary. However, by using recursion (i.e., feedback) to make the output a function of both output and input, a transfer function that is more general and more economically achieved (but more difficult to design and potentially unstable) becomes possible.

Recursion may be achieved by feeding forward (to the output) and back (to the input) from each tap point via individual coefficient multiplications (Figure 6.11a). A more elegant scheme uses step by step recursion, employing cells having identical form, in which each cell has two inputs—a direct and a recursion input, a delay, two multiply-and-adds, and two outputs, one going forward—the other going backward—forming a *lattice* structure (b).

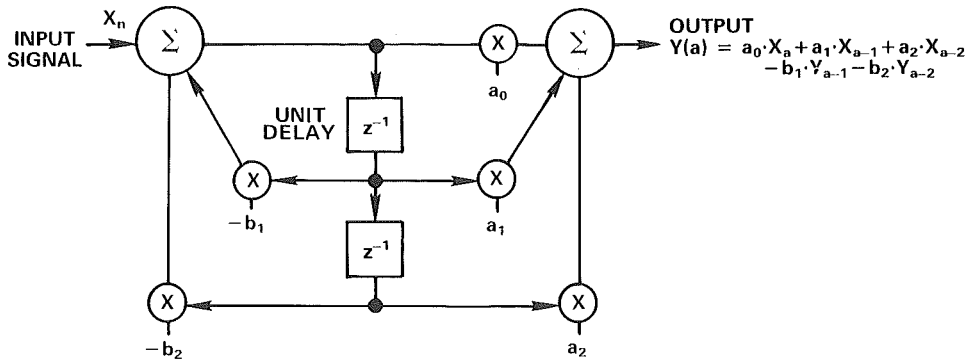
6.2.2 CONCLUSION

Though it has been limited in scope, we hope that this section has provided the reader with an awareness of the power of digital techniques in signal processing, just through the use of the delay-line storage model. There are many more processing tricks available, if one is open to considering digital and analog, hardware and software, alone or in combination.

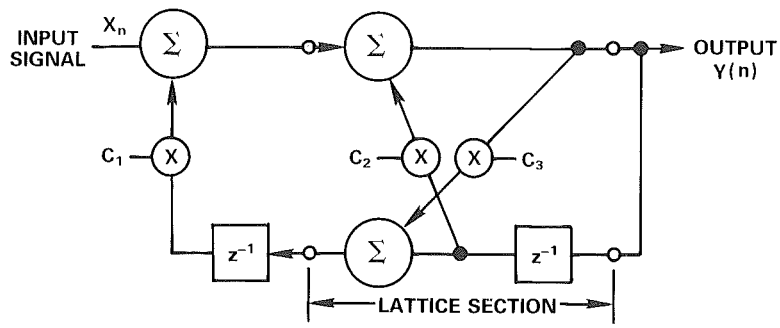
For use between the input a/d converter and the output DAC or display, the growing availability of digital components (see Chapter 21) of high complexity, increasing speed, and low cost (e.g., IC multiplier-accumulators and array processors), plus the availability of large amounts of memory and the possibility of overall control of the processing by CPUs and stored commands, has

*i.e., if the input is doubled, the corresponding output will be doubled.

[3]For an example, see Windsor and Toldalagi, "Digital FIR Filters without Tears", *Analog Dialogue* 17-2, 1983.



a. Feedback structure.



b. Lattice cell.

Figure 6.11. Two types of recursive filters.

made the outlook for analog waveform synthesis, analysis, and processing by digital techniques extremely bright, whatever the source. Speech, music, noise, gas chromatographs, electroencephalograms, image data from medical scans—whether NMR, ultrasound, or X-Ray—and mechanical vibrations are just a few.

6.3 CATHODE-RAY-TUBE DISPLAYS

In the industrial and scientific world, the close association of computer and cathode-ray tube provides an unparalleled method for speedy access to stored and real-time data, programs, and graphics. At the same time, it affords the opportunity for interactive dialogue with the computer, and through it, the system under test or control, for the purpose of actually instructing the computer or controlling real or simulated processes in real time. The recent growth of electron-beam recording (on film negative) poses a serious challenge to the centuries-old tradition of typesetting, while the ability to use

computer power to adapt data to the needs of the human operator prior to presentation makes the computer-CRT display an all but indispensable combination.

While systems do exist for the sole purpose of display (i.e., monitors), the more general application of displays is in connection with data-acquisition systems and interactive systems involving computers. Some such systems deal with data that is purely alphanumeric by nature (e.g., accounting and word-processing systems). Others maintain contact with real-world physical variables through the use of a/d and d/a converters, either directly on-line in real time, or through recapitulation of data already captured. This category includes storage oscilloscopes and graphic-display scopes.

Whatever the display's purpose—or the source of the data—many cathode ray displays involve the use of D/A converters for generating sweeps, characters, and vectors, for positioning and intensification, relying on their inherent linearity, reproducibility, and controllability by entirely digital sources of command.

Since we are concerned here primarily with display systems that employ converters—with particular emphasis on the way they are used and the factors of importance in selecting and using them—the number of systems chosen will be limited and system descriptions will be brief.

The graphic-display oscilloscope can be found in increasingly widespread use—doubling every few years—in a growing variety of applications that were just wild dreams (if that) just a few years ago. Today's displays are characterized by high resolution, the promiscuous use of color, and a wide range of hardware and software options.

In general, a cathode-ray display system consists of a display-processing function and the CRT hardware, usually integrated into a single package. The processing function includes buffer storage to hold the information to be presented for update, the instructions for presenting it, the signals needed to activate the display elements, synchronization signals, and the digital-to-analog processing hardware; it may include a *refresh memory*. The CRT hardware comprises power supplies, CRT, circuitry for beam positioning intensification, nonlinearity correction, and focus.

Representative display techniques include:

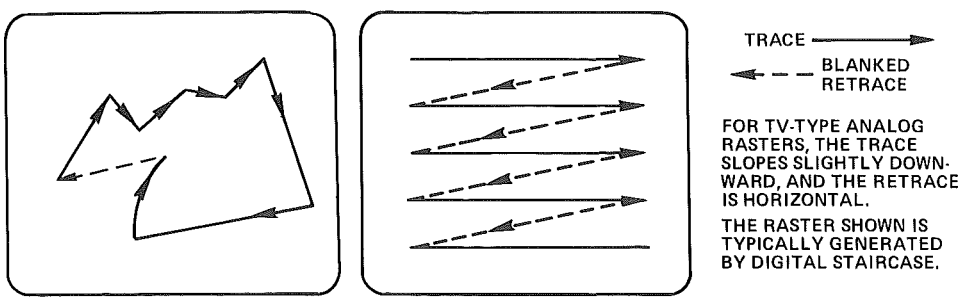
TV raster (picture and graphic displays)
Stored-character display, e.g., Monoscope (alphanumerics)
Dot-matrix (alphanumerics)
Cursive: stroke and vector generators (alphanumerics and graphics)
Rotating (PPI)

Reams of material have been written in recent years, debating the pros and cons of various CRT display systems. There is no "best" system for all pur-

poses; they must be compared in terms of their advantages and disadvantages for specific applications.

Even the methods of deflecting the electron beam (or beams, for multicolor displays) have been subject to discussion. Systems with electromagnetic deflection use a magnetic yoke around the neck of the CRT to deflect the electron stream; systems with electrostatic deflection use voltages applied to sets of deflection plates built into the tube to steer the beam. Electromagnetic systems are slower and require more power than electrostatic systems, but they are cheaper and are widely used in TV-type raster-scan displays. Some systems use both—for example, electron-beam systems for integrated-circuit manufacture.

With computer processing, memory, and software widely available at low cost, computer-controlled displays have become popular in two basic forms: the *vector-refresh* (*alias* random-scan, calligraphic, stroke-writing, or directed-beam) display, which forms X-Y plots of digitally determined points or digitally programmed line-segments of random length and direction (in similar manner to oscilloscopes in the X-Y plot mode), and the TV-like *raster scan*, which places a large number of closely spaced dots of variable illumination—and color—along a raster of closely spaced horizontal lines every 1/30 or 1/60 of a second. Although usage of both techniques is growing, raster-scan is the more popular and growing at a much faster rate. Simplified traces characteristic of the two types of display are shown in Figure 6.12.



a. Randomly programmable vector-scan beam path.

b. Repetitive raster-scan beam path.

Figure 6.12. Display traces compared.

Vector-scan systems move the beam only to those portions of the screen forming the illuminated portion of the pattern, while the beam that illuminates the raster scans the entire screen, irrespective of the number and location of pixels to be brightened. In most applications, raster-scan displays use electromagnetic deflection (an inheritance from TV circuitry), while vector types use elec-

trostatic deflection. The former has the advantages of simplicity and low cost; the latter requires less power, provides better resolution, and can display data changes with precision. Raster-scan systems are faster if large quantities of data are changing from frame to frame, while vector scan is often the choice where definition (i.e., precision and resolution) is important.

6.3.1 BASIC SYSTEM

Figure 6.13 shows the generalized system outline for an installation capable of accepting, processing, storing, and displaying information on a CRT screen. Conventional business data-processing systems do not normally in-

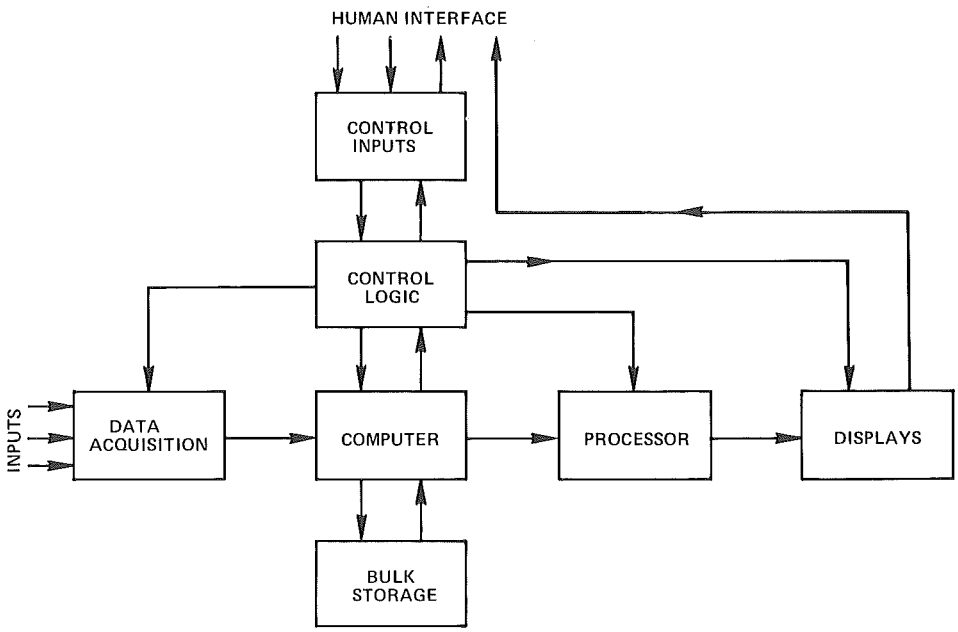


Figure 6.13. Display system outline.

volve sensors and A/D interfaces—but do involve other forms of peripheral data input. A computer game, employing a joystick for cursor position control, and an air traffic control system, based on radar data, are examples of ways CRT displays are used for interactive handling and presentation of information derived from the analog domain.

Further ingredients of the generalized display system are quite straightforward. The manual controls provide human interface, enabling the operator to call for a specific picture (or portion of a picture), to enter new information into the system, to command new modes of operation, and to initiate different data-processing and display functions. Bulk storage forms part of the data-

processing capability; further auxiliary storage is often used for display refreshing at high speed to avoid annoying flicker. Control logic interfaces between computer data and the various peripheral devices, including displays, memories, communications links, the human operator, data-acquisition circuits, etc.

Driven to ever-higher scanning speeds to obtain increased resolution without flicker, the displays require extremely fast information transfer. For example, in a display with 1280×1024 fine structure, there are 1.31×10^6 picture elements (*pixels*). At 60 updates per second, each pixel must be displayed in less than 13 ns, if every data point in the field must be capable of being plotted, as is the case in raster graphics.

6.3.2 USES OF D/A CONVERTERS IN DISPLAYS

Raster Displays

Raster-scan graphic displays are very much like television pictures (and, in fact, often use TV hardware): as each vertical sweep scans linearly down the face of the tube, repeating its course every 1/30 or 1/60 of a second, the electron beam is repeatedly, rapidly, and linearly drawn across the face of the tube, forming a large number of equally spaced horizontal scans (lines), during which the signal information is modulating the intensity input of the cathode-ray tube. The major departure from TV is that, in high-resolution displays, there are many more horizontal lines, and more points per line. As in TV, the scans are synchronized, so that points brightened at the same time after the start of each horizontal sweep are directly above one another, and—if recurring on successive sweeps—will form a vertical line.

Because all points on the raster are scanned on each full cycle, the raster-scan cannot be as fast and sharp as the calligraphic display, especially for plotting simple figures; however the synchronization and standard nature of raster displays make it easier to obtain high-resolution shading, using the intensity gray scale, and to obtain a very large variety of color mixtures and hues, at low hardware cost.

One fast d/a converter channel provides intensity modulation for each electron gun: a single one for monochrome displays, three for color. Figure 6.14 is a block diagram of a typical computer-controlled raster-scan monochrome graphic display system.

A typical system comprises a MOS random-access memory buffer for storing display data in digital form, one or more memory controllers for managing the updating of the display and controlling the refresh cycle of the CRT, and a programmable microprocessor for generating display graphics and manipulating the image. The entire system operates as an intelligent peripheral

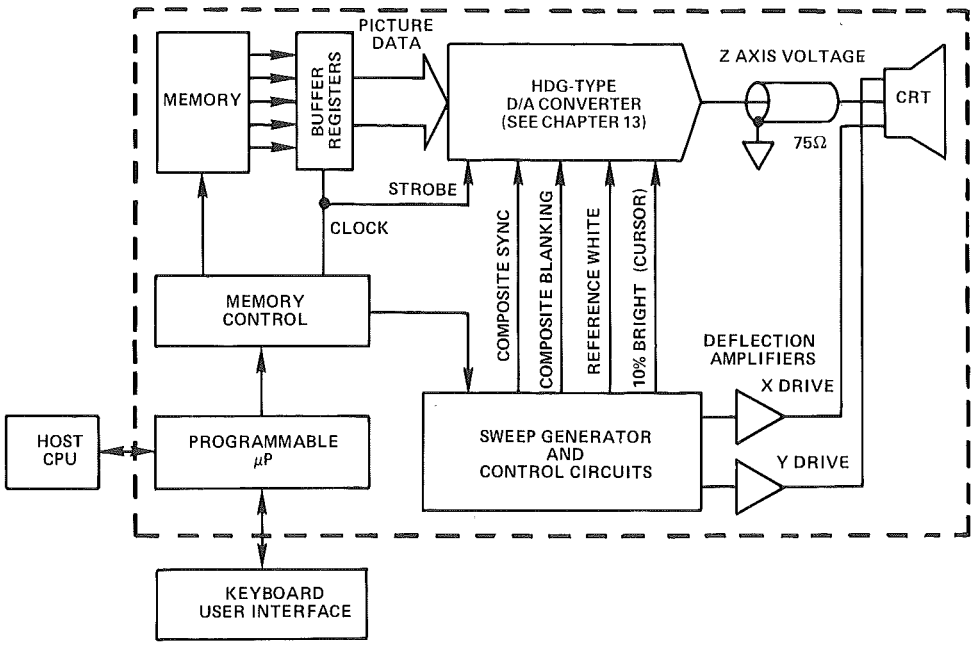


Figure 6.14. Raster-scan display system.

to a host computer; most of the processing associated with image and graphic display is down-loaded to the graphic subsystem.

If the picture resolution is specified as 1,024 × 1,024, there are 1,024 horizontal lines, each having 1,024 independent dots, and each dot has its own programmable intensity level; thus, there are 1,048,576 independent pixels. If the picture on the CRT is to be refreshed 60 times per second, then a new pixel must appear on the screen at least every 15.8 nanoseconds (exclusive of "overhead time" required for vertical and horizontal blanking during the sweep retrace portions of the cycle).

The d/a converter controls the Z-axis of the CRT, to modulate the brightness of the raster-scan beam. For the example mentioned above, the DAC must be capable of being updated at the pixel rate corresponding to 15.8 ns; it should be capable of settling to a new value in less than 10 ns. If a white dot is being plotted on a black background, the DAC's output must make a full-scale transition between adjacent pixels. The resolution of the DAC determines the number of finite intensity levels available. Typical DACs for this purpose have resolutions ranging from 4 to 8 bits, corresponding to from 16 to 256 levels of gray scale. For color displays, three memories and three DAC channels are required, one for each color gun of the CRT (red, green, blue). Triple DACs in monolithic form are becoming available for the purpose (e.g., the AD9702).

In addition to the programmed levels during the visible portion of the sweep, the electron beam must be blanked during the retrace. It is also useful to have an extra-bright level for the cursor in interactive applications. Besides a standard range of code-controlled intensity output levels, from reference black to reference white, special-purpose display DACs have control inputs that produce additional voltages to furnish standard levels of "blacker-than-black" for blanking and whiter-than-white for cursors.

Figure 6.15 shows the standard composite intensity waveform over 1 ½ cycles of the horizontal sweep. The controlled range of the DAC's full scale span (−643 mV) is from reference white (−71 mV) to reference black (−714 mV).

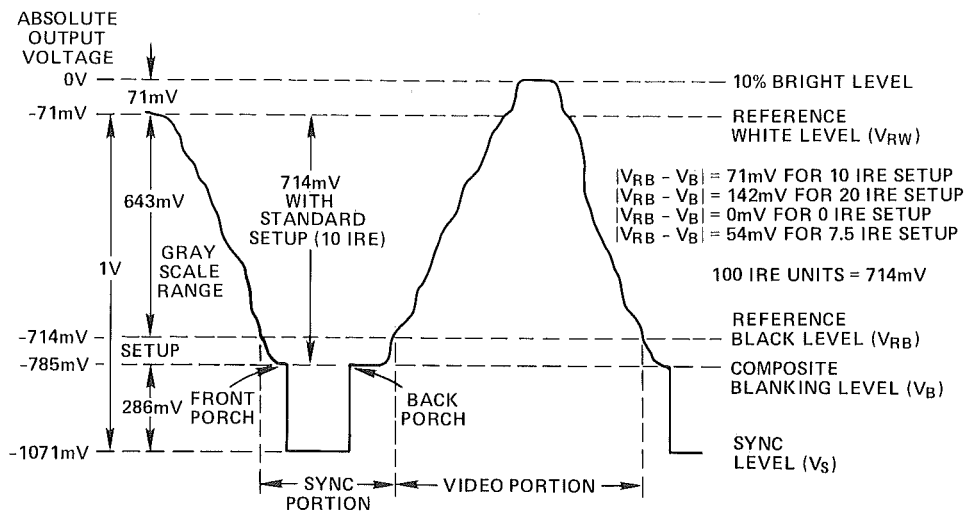


Figure 6.15. Composite DAC output waveform in raster-scan display system.

In the illustration, the intensity over the sweep interval is shown varying from full white to full black. At the beginning of the synchronized retrace portion of the cycle, the intensity signal drops to the blacker-than-black "front porch" (−785 mV), and then to the extreme black level (−1071 mV) during the horizontal retrace. As the next sweep starts, the intensity returns to the "back porch" (−785 mV) and, as the first element of the picture is triggered, to the controlled range of the DAC. During this scan, the cursor is displayed at the 10% "brighter than white" level (0 mV).

Sweeps may be produced either by analog ramps (integrating currents through capacitors) or by converting staircase outputs from digital counters to analog via DACs. D/A converters are especially well-suited to vertical sweeps, for a number of reasons:

- Timing, controlled by a clock and logic, is quite precise and uniform.
- Lines are horizontal (analog sweeps have slight tilt).
- Line-spacing uniformity depends on linearity, while maximum number of lines depends on DAC resolution. DACs having 10-bit-or-more resolution (1024 + lines) and 12-bit linearity (0.0125% linearity error) are readily available. In electron-beam recording, a 16-bit DAC can provide 4096 lines with less than 5% spacing error.
- DAC full-scale switching transients are blanked because they occur during the horizontal retrace interval.

In Section 5.5, a counter-driven D/A converter was suggested as a sawtooth sweep generator. When used for displays, such schemes can provide highly-repeatable, controllable, and linear sweeps of arbitrary resolution and accuracy.

For horizontal sweeps, the requirements on DACs are more severe, and analog sweeps win the cost tradeoff in many applications. For example, to resolve 500 points per line, at 500 lines per frame, at a 60-Hz frame rate, requires that each digital horizontal step settle well within 100ns, and that there be no "glitches," distorting the scan at major transitions and producing vertical stripes.

Vector Graphics

The general objective in vector graphic displays is to provide a flicker-less high-resolution presentation of numerical, line-drawing, or pictorial information.

The usual problem is to start with the spot at a point having a given set of coordinates (which may be any point arrived at in the course of plotting the display, or it may be the starting point of a pattern), and plot a line to another point. A variety of methods have been used employing such analog techniques as modulating and summing ramps, performing integrations, etc. These methods, although using complex analog circuitry, are economical of digital data and resolution.

However, as fast deglitched d/a converters with high resolutions (12 bits and more) have become available, and as buffer memory becomes denser and cheaper, there is a trend towards defining all the data digitally and using point-by-point plotting.

Figure 6.16 is a block diagram of a typical vector-scan display system that plots point-by-point, using 12-bit DACs for the X and Y axes, and a fast DAC with coarser resolution to modulate the Z-axis (intensity). To obtain the perception of continuous lines when the display consists of discrete points, it is essential to use high-resolution d/a converters to drive the X- and Y-axis inputs. For example, a pair of 12-bit converters will provide a display of

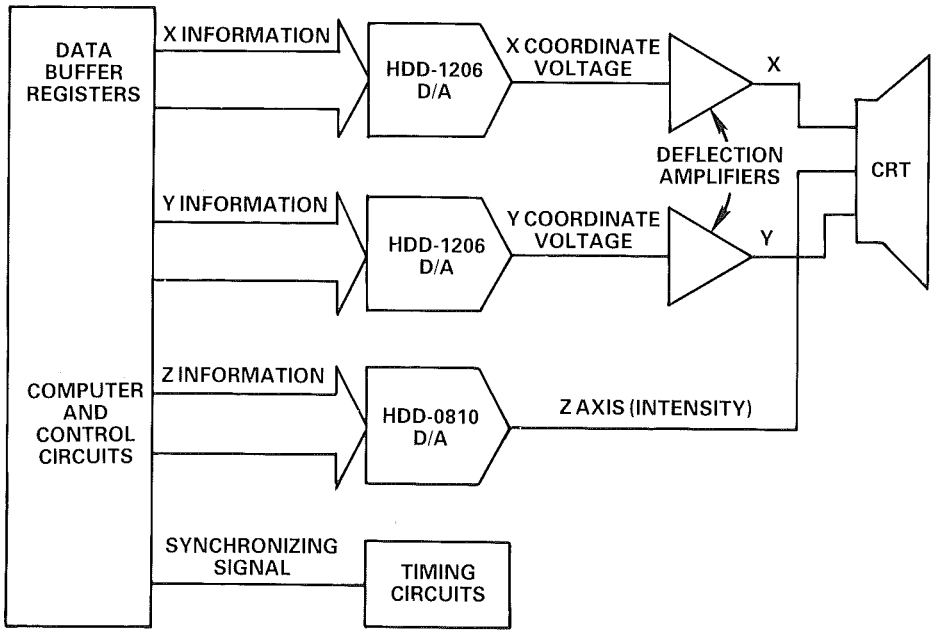


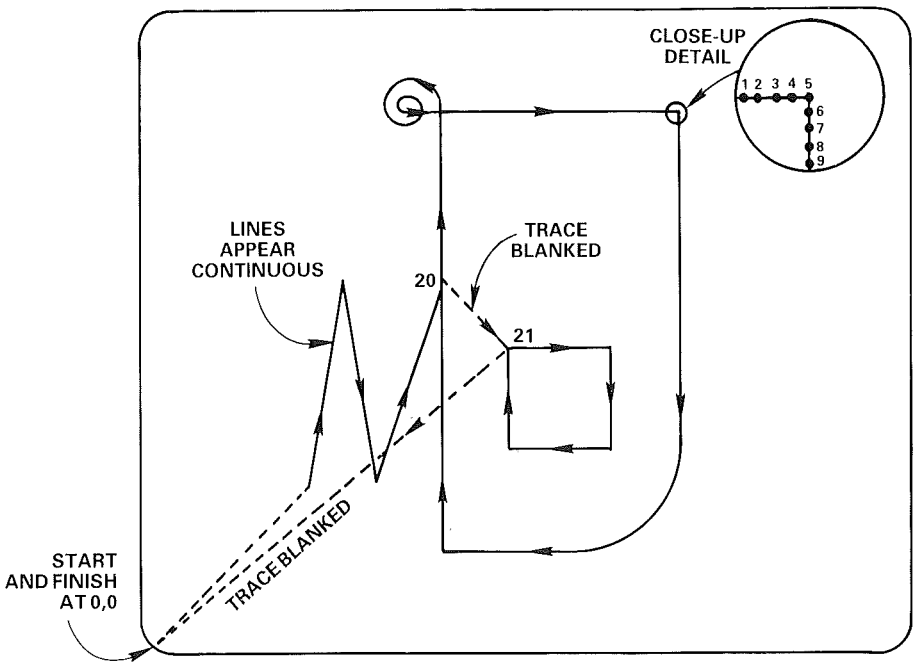
Figure 6.16. Typical vector display system.

4,096 × 4,096 bits, or about 16.8 million pixels. With a 21″ screen, resolution of about 0.1 mm would be available.

The high density of available pixels makes it mandatory that only high-resolution, high-quality "deglitched" DACs be used for plotting the digital data stored in memory. Any aberration in the DAC output, static or dynamic, can introduce an error in the output voltage and erroneously position the plotted point.

Figure 6.17 shows a simplified randomly programmable vector-scan beam path. The X-coordinate DAC moves the beam left-and-right; the Y-coordinate DAC moves it up-and-down. The beam is allowed to provide continuity between connected points, but it is blanked when it skips over a dark area. Errors in positioning in the X or Y direction produce lines with distorting lumps. D/A converter technology today is sufficiently advanced to ensure that problems with dc linearity and monotonicity are essentially nonexistent; so dc characteristics are not a significant source of errors in vector-scan displays.

However, besides high dc resolution, the d/a converters need a few other features. First and foremost is speed. The faster the DAC, the more points that can be illuminated within a given scan (usually 1/30 to 1/60 seconds, to avoid a flickering appearance). A converter—like the HDD-1206—with 60-nanosecond settling time for one-bit changes (while plotting a line) is compatible with a 6-MHz refresh rate; and only 2 μs are required for an accurate full-



CODES AT INSET

POINT	X	Y
1	1010 1010 0001	1110 0100 1111
2	1010 1010 0010	1110 0100 1111
3	1010 1010 0011	1110 0100 1111
4	1010 1010 0100	1110 0100 1111
5	1010 1010 0101	1110 0100 1111
6	1010 1010 0101	1110 0100 1110
7	1010 1010 0101	1110 0100 1101
8	1010 1010 0101	1110 0100 1100
9	1010 1010 0101	1110 0100 1011

CODE AT JUMP

20	0110 0100 0010	1001 0100 0011
21	0111 1010 0110	0111 1001 0100

Figure 6.17. Randomly programmable vector-scan beam path.

scale jump (less time for shorter jumps). Effective plotting strategies would call for many connected points and only short jumps, when feasible.

But high speed and high resolution are useless if the transition from point to point is not clean. A major problem in fast high-resolution DACs is the discontinuity, or "glitch" (Section 3.7), that can occur at major transitions, for example, in the one-bit transition from 0111 1111 1111 to 1000 0000 0000. If the less significant bits turn off faster than the MSB turns on, the output will swing rapidly towards zero, then rapidly swing back towards the original-level-plus-one-bit, causing the trace to swing wildly over the face of the CRT in the course of making that small change. Even at less-significant transitions, highly visible perturbations can occur.

Since the glitch is a code-sensitive nonlinear phenomenon, it cannot be simply filtered by linear techniques. It must first be minimized in duration and amplitude by designing the switching to be as symmetrical as possible; then the residual glitch can be eliminated by using a track-hold output amplifier circuit. When a new value of digital input is latched in, the output circuit switches to *hold* to retain the previous value until a time when the glitch can be expected to have settled out, then switches back to *track* to acquire the new value.

The "before" and "after" photographs in Figure 6.18 show the effects of glitches in a display and the great improvement that can be effected by de-glitching. (Photos courtesy of The Foxboro Company.)

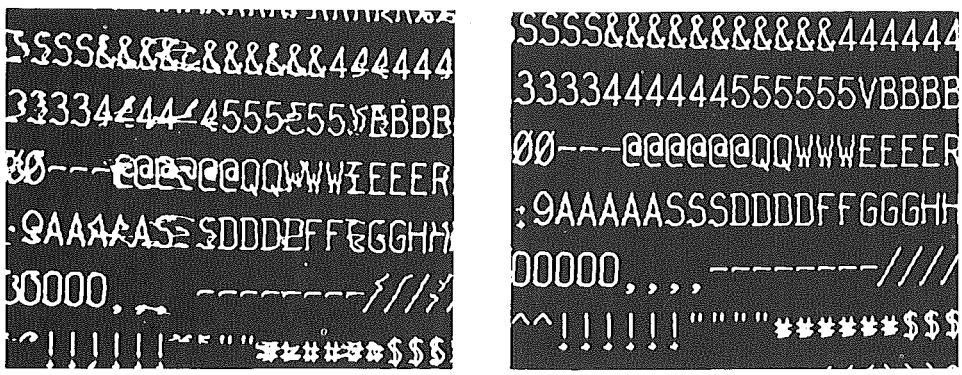


Figure 6.18. Effect of "glitches" on a display.

Dot-Matrix Displays

In vector graphics, elements that are repeated—for example, alphanumeric characters—need not have every point's complete address stored and traced out by the master X-Y DACs. Instead, a stored-character dot-matrix can be employed.

Each character is represented by a matrix of points, e.g., 4×7, with each point that is defined as part of the character intensified, by (for example) a character trace. The X and Y coordinates of each point of the character are located at addresses in two ROM's; the point is addressed by a word consisting of a format code for the character (e.g., ASCII) and a number from a counter indicating the order of the point in the writing sequence (i.e., i, in x_i, y_i).

In a typical system using this presentation (Figure 6.19), the outputs of two high-speed, low-resolution character DAC's (X and Y) are summed with the outputs of the main DACs. The main DACs establish the X and Y coordinates of the position of an index point on the character. The second set of DACs produce a sequential set of outputs that rapidly move the spot from one point to the next, dwell, and move on, until the character has been traced out.

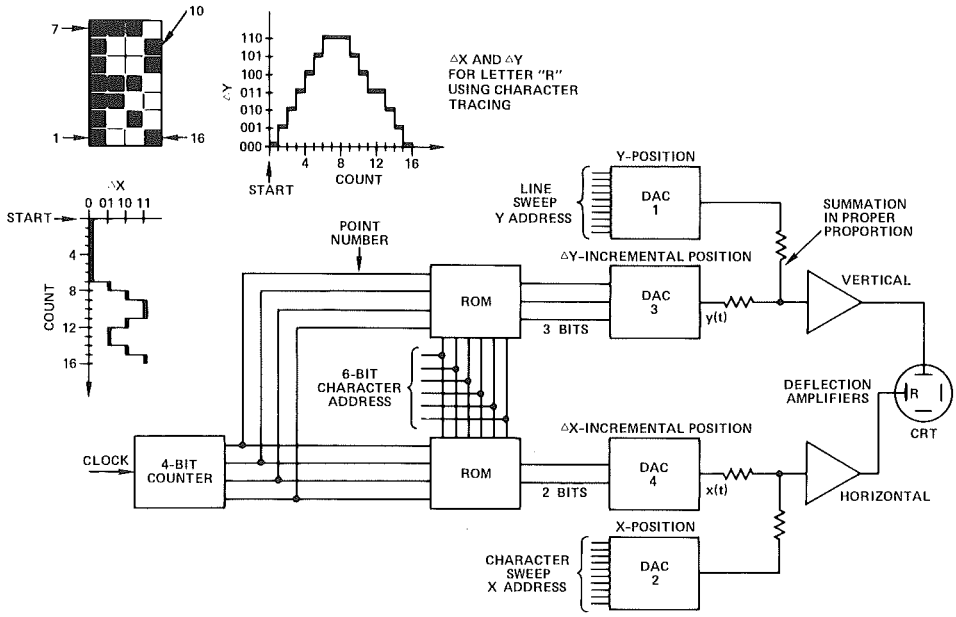


Figure 6.19. Dot-matrix display scheme.

An important advantage of this scheme is that the DACs that produce the characters need only have fast response (generally considerably faster than the main DACs), with very modest resolution and accuracy. In addition, the refresh memory needs to store only the character codes, rather than absolute point locations for each character.

Although 2 and 3 bits served to display the character adequately in the example, the D/A converter may have many more bits available for handling other forms of additive input. It is important to note, though, that the accuracy and resolution of the positioning DAC must be such that its errors are less than the relatively weighted value (taking differing scaling into account) of the least significant bits of DACs whose outputs are summed. Otherwise, overlapping or uneven spacing may result.

6.4 COMMERCE, INDUSTRY, AND ELSEWHERE

Because A/D and D/A converters were originally developed as computer interfacing devices, used primarily for getting data into and out of digital computers, the casual observer still tends to associate them with computer application alone. In reality, as Chapter 5 has demonstrated, A/D and D/A converters, as components, have followed the operational amplifier out of the computer laboratory and into the industrial world-at-large. But then, too, so has the computer, also as a component!

The reader who has arrived at this point (after presumably reading all of the material in Part One) has been exposed to a large variety of circuit configura-

tions and application suggestions. It would not have been difficult to have noticed that some of the configurations looked more-or-less alike, though offered from somewhat different viewpoints.

In this section, closing the chapter and Part One, we will show just a few end-user applications, with the descriptive emphasis more on what they *accomplish*, rather than on how their circuits go together. The reader will not find anything especially different, from the circuit point of view, but it will serve as a microcosmic glimpse of end applications of conversion devices in the workaday world.

The applications include:

Electronic instruments
Medical imaging
Industrial automation
Oil-Well Monitoring

Electronic Instruments

Designers of test and measurement instruments, increasingly using automated measurement techniques that employ microprocessors, are finding a/d and d/a converters essential for direct measurements, setting gains, calibrating, and performing comparisons against standards. Electronic measurement techniques are used for accurately measuring virtually every conceivable parameter, including temperature, light, chemical composition, and pressure, including of course the familiar electrical parameters—as well as the performance characteristics of electronic equipment.

In performing measurements, an instrument's sources of error must, as a rule, be significantly less than other error sources in the measurement, or—for testing—in the device being tested. The concern for accuracy becomes particularly essential in the case of key components located within the critical measurement path.

A typical application where converter accuracy is a key to the desired measurement accuracy is found in sweep oscillators for wide-range testing of radio-frequency equipment. For example, Figure 6.20 is a simplified block diagram of Hewlett Packard's Model 8340A synthesized sweeper, a μP-controlled sweep generator that uses frequency synthesis techniques to generate signals ranging in frequency from 10 MHz to 26.5 GHz. Its output frequency can be swept over ranges up to the full frequency range of the instrument, making it adaptable to a wide variety of applications.

The value of the YIG-tuned output frequency range and offset, determined by the microprocessor, is established by a tuning voltage derived from an analog signal from a sweep-generator board and applied to the frequency generator. At the heart of that board (Figure 6.21) are three d/a converters—a 10-bit converter to control the gain of an integrator-generated sweep ramp

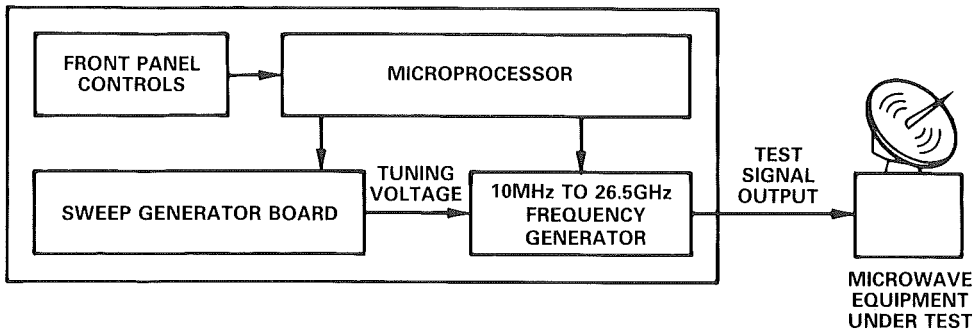


Figure 6.20. Synthesized sweeper using DACs to control the instrument's tuning voltage.

(×8 and ×64 gain switching provide an overall gain range of about 64,000); and a 12-bit (master) and an 8-bit (vernier) converter, used together—with 2-bit overlap—to provide an offset range of 262,000 values.

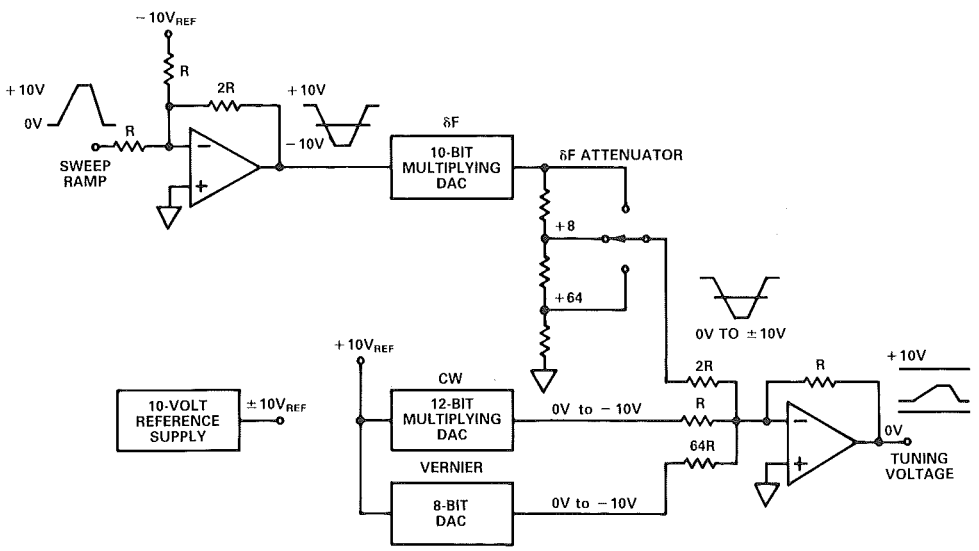


Figure 6.21. Sweep scaling and offset circuits condition the sweep ramp before it is sent to the frequency generator.

Medical Imaging

Since the time that German physicist Wilhelm Conrad Roentgen first performed his historic x-ray experiments in the late 1800s, medical innovators have sought continuously to increase the effectiveness of medical imaging methods for clinical diagnosis and therapy. Recent advances in imaging tech-

nology have provided diagnostic tools that are faster, safer, and more effective than ever before. In the area of x-ray technology, one of the most impressive recent developments is computerized axial tomography (CAT).

Computerized axial tomography is a technique for taking cross-sectional x-ray pictures of the body in planes perpendicular to the body's vertical axis.

Standard x-ray picture techniques, using photographic film, can diffuse x-ray energy throughout large portions of the body, including areas outside the area of immediate interest. The cumulative dosage over many pictures can be harmful to the patient. In addition, it is often difficult to interpret the pictures, because structures casting heavy shadows may be aligned with and block the images of other tissues that are of interest. This situation can be to some extent ameliorated if pictures could be taken from several different angles and then superimposed, eliminating redundant information, but the many pictures sometimes required could call for intolerable dosages of x-ray energy.

CAT scanning is a system approach that accomplishes the superposition of multiple images electronically, with considerably less irradiation of the patient—especially in portions of the body outside the area of interest. The major elements of the system are shown in Figure 6.22.

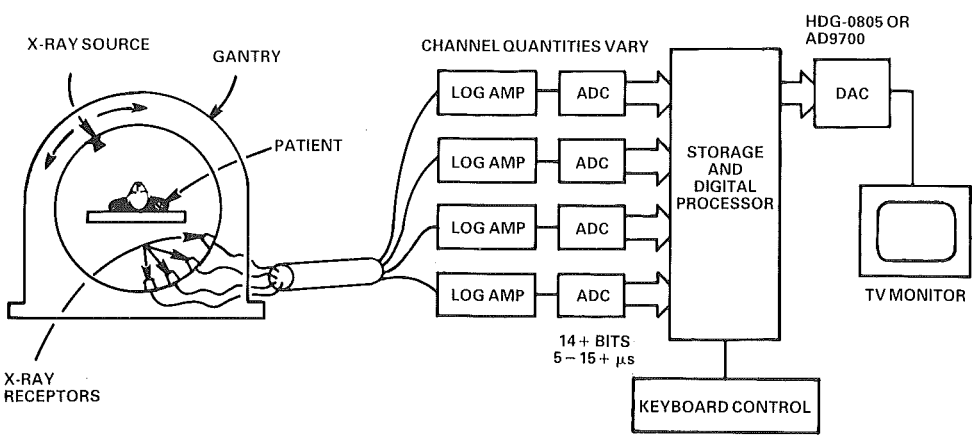


Figure 6.22. A/D conversion is the key to the effectiveness of CAT scanners. Once in digital form, the data can be processed to provide high-resolution pictures of cross-sections of internal body structures.

The x-ray source is mounted on a rotating cylinder within a large gantry, and the patient's body is centred within the cylinder. This positioning allows each successive exposure, measured with a set of sensitive x-ray receptors, to obtain data that can be accumulated and processed to produce a cross-sectional picture, or "slice," of the patient.

The scanner takes a picture in its initial position; the rotating cylinder, with source and receptors, is rapidly repositioned by 2° or 3°; a second picture is taken, and the process continues for a full 180°, requiring from 1 to 5 seconds

for the complete cycle. After each "slice," the movable bed on which the patient is lying is incrementally advanced to a new position in the gantry. This allows multiple cross-section views to be stacked to form a three-dimensional picture of the part of the body being examined.

The analog data—collected by the receptors—for each picture are applied, via logarithmic amplifiers, to analog-to-digital converters. These converters often have 14+ bits of resolution and perform the digitizing function in 5 to 15 microseconds. The number of bits is dictated by the need for a wide dynamic range in order to observe small differences in image density that may be important in diagnosis.

After each 180° scan, the signal processor assembles the completed picture; it starts by noting the intercept points of the most dense tissue at each of the stopping points in the scan. Cross-correlation of these points allows the computer to assemble an extremely accurate picture of the interior of the body.

In addition to CAT scanning and various other approaches to medical imaging, there are many other occasions for data acquisition and conversion of real-world signals in health-care equipment, including patient-monitoring equipment, blood and body-fluid analysis systems, and other systems tht require measurements of such variables as body temperature, blood pressure, tissue structure, body chemical composition, and fluid flow.

Industrial Automation

Factory automation systems link automated, computer-controlled production- and materials-handling equipment with sophisticated communication networks. The continued viability of major segments of manufacturing industries worldwide will be dependent on the further development and implementation of factory automation products that assist in increasing productivity and quality, as well as lessening hazards in the workplace.

The current high level of interest in robotics technology is a direct result of the increased industrial manufacturing productivity it promises. Worldwide sales of industrial robots were expected to grow more than sixfold between 1983 and 1990.

Industrial robots are "intelligent" machines capable of independently performing various types of programmable factory tasks. They simplify many repetitive operations, such as sorting, inspection, and assembly, by putting those tasks under microprocessor control.

Typically, robots are required to perform two major types of tasks: those involving some kind of sensing, and those involving some form of locomotion or actuation. Whether the robot arm holds a drill, an arc welder, or a spray painter, the basic operation is the same: the robot must sense the object it needs to work on, it must locomote to that object, and it must quickly and efficaciously drill the desired hole, execute the desired weld, or coat the ap-

propriate surface with a desired thickness of paint. Throughout this process, there is a feedback loop, which enables the robot to check its performance against the specified parameters. To effectively interface with its factory environment, the robot must operate with a very high degree of precision.

For example, the Mitsubishi Electric RW-1A is an arc-welding robot that has virtually the same manipulative ability with arc-welding tools as the human hand. In addition to the problem of positioning the arm, the robot must also maintain a desired profile of welding current. The welding current is sensed and converted to digital by a welding current sensor (Figure 6.23), using a

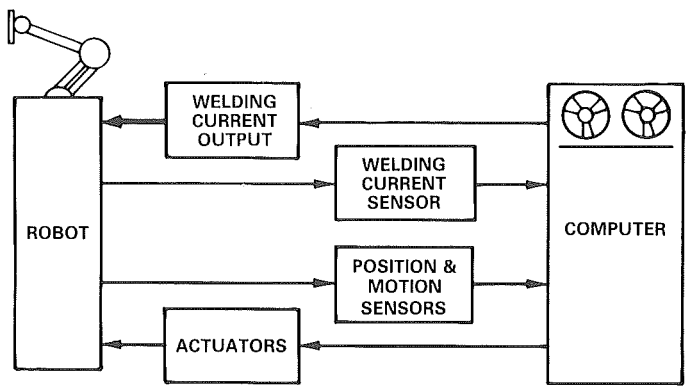


Figure 6.23. A 12-bit integrated-circuit ADC is used in the welding current sensor in monitoring and control of the robot's welding arm.

12-bit ADC, and the computer digests this feedback information and provides the adjustments necessary to maintain the specified values of current.

In addition to robotics, aspects of industrial automation providing opportunities for conversion devices include batch and continuous process control, and automatic inspection and test equipment.

Oil-Well Monitoring

High energy costs and growing concerns over the limits of the world's energy sources have led to the increasing use of automation as a means of using existing resources more efficiently and economically. This is especially true in the petroleum industry, where automation is being applied to all phases of production—recovery, refining, and distribution.

"Recovery" refers to the means by which oil and gas are brought to the surface once a producing well has been drilled. In most fields, the earlier wells produce by natural flow—that is, the initial pressure within the well is sufficient to cause the oil to flow to the surface without assistance. However, heavy crude oil does not flow readily at normal subterranean reservoir temperatures, and, as more oil is pumped from the ground, the pressure and natural flow of the well decrease. Natural recovery techniques are no longer effective for

many of the mature oil fields in the United States; unenhanced processes typically recover only 15-30% of the oil in place.

The industry's push for increased production has led to enhanced oil recovery (EOR) projects, in which reservoir temperatures are raised through the use of steam injection. EOR is quite costly, so it is essential that field equipment be capable of effectively monitoring the pressure, temperature, and flow rates of the injected steam and the resulting fluids, to achieve the most efficient use of available material and economic resources. In the case of remote well sites, field equipment must also be capable of transmitting the data to a central location for analysis and control of the recovery operation.

As an example of this, a producing division of Texaco implemented remote monitoring on an EOR project involving 13 sites located more than 1 mile from the central computer. They wanted equipment at each well that could measure a broad range of critical parameters, and then convert the analog data into digital form for radio transmission back to the central computer. The means of performing the multiple-channel data acquisition—and remote control—was an integrated single-board measurement-and-control subsystem (see Table 4.1 and sections 4.1.6 and 4.3.3).

As Figure 6.24 shows, at each remote site, a μMAC-4000 accepts real-world signals from various sensors, such as thermocouples, flow meters, and pressure gages. These signals are filtered, amplified, and otherwise conditioned, then converted to a format which may be transmitted. Upon request from the host computer, the μMAC-4000 sends stored data via radio waves to the central location. Here, the data undergoes additional processing and analysis before the information is presented by the computer in printout form. If an engineer's review of the printout suggests that an adjustment in the EOR process is desirable, instructions can then be sent from the central computer back to the μMAC-4000, which will issue control commands to the appropriate valves and motors.

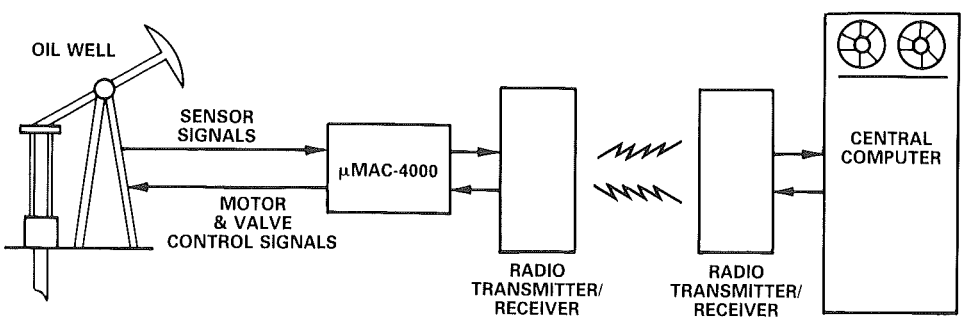


Figure 6.24. An integrated measurement-and-control subsystem provides effective monitoring of the enhanced oil-recovery process, ensuring efficient resource allocation.

The need for computer-aided measurement-and-control solutions, which involve conversion, in all levels of integration—from signal conditioning and conversion to complete stand-alone measurement-and-control systems, replete with field sensor connections and sophisticated software—exists wherever there is a need for an interface between the real-world process being measured and the information system. Applications for such subsystems include, not only energy conservation and development, but industrial process control, machine control, product test, laboratory research and development, and much more.